University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

شبكات الحاسوب Multimedia Networks

Teaching by: L. Wisam Mahmood
م. وسام محمود
Prepared by: Dr. mohamad nataq
أ.م.د. محمد ناطق

cs.uotechnology.edu.iq

1. Multimedia Networks

Multimedia networks refer to computer networks that are designed to handle different types of media, such as text, graphics, audio, and video. These networks can transmit and receive multimedia data in real-time or non-real-time modes, and provide users with a wide range of services, such as web browsing, email, instant messaging, video conferencing, and online gaming.

Multimedia networks are typically composed of several components, such as servers, routers, switches, and clients. Servers store and distribute multimedia content, while routers and switches manage the flow of data between different network segments. Clients are the end-user devices, such as computers, smartphones, and tablets, that access multimedia content and services.

To ensure the efficient transmission and delivery of multimedia data, multimedia networks use different protocols and technologies, such as TCP/IP, HTTP, FTP, SMTP, DNS, and RTP. These protocols help to standardize the way multimedia data is transmitted, ensuring compatibility between different devices and platforms.

Overall, multimedia networks play an important role in enabling users to access and share different types of media content across different devices and platforms, making it an essential component of modern communication and entertainment.

1.1. Multimedia Networking Challenges

Multimedia networking faces several challenges due to the diverse nature of multimedia data and the complexity of modern networks. Here are some of the challenges faced by multimedia networking:

1. Bandwidth limitations: Multimedia data, such as high-resolution images and video, require high bandwidth for transmission. However, network bandwidth is

often limited, which makes it challenging to deliver multimedia content with acceptable quality.

2. Quality of Service (QoS) requirements: Multimedia data has strict QoS requirements in terms of delay, jitter, and packet loss. Achieving these requirements is challenging, especially in networks with varying traffic loads.
3. Network heterogeneity: Modern networks are composed of heterogeneous devices and technologies, such as wired and wireless networks, different operating systems, and different network protocols. This heterogeneity makes it challenging to develop a multimedia networking solution that works seamlessly across different devices and networks.
4. Security: Multimedia data is often sensitive and requires secure transmission. However, securing multimedia transmission is challenging due to the complexity of multimedia data and the variety of security threats.
5. Interoperability: Multimedia networking solutions need to work with various multimedia formats and protocols to provide seamless transmission across different devices and networks. Achieving interoperability between different multimedia formats and protocols is a significant challenge.
6. Content protection: Multimedia content providers often require that their content be protected against piracy and unauthorized access. However, protecting multimedia content is challenging due to the ease of copying and distributing digital content.
7. Scalability: As multimedia data becomes more prevalent, the demand for scalable multimedia networking solutions is increasing. Providing scalable multimedia networking solutions that can handle a large number of users and multimedia data streams is a significant challenge.

Overall, multimedia networking faces several challenges that require innovative solutions to provide reliable and high-quality multimedia transmission over different networks and devices.

1.2. Dimensions of Multimedia Communication

Multimedia communication is a broad area that involves different types of media and technologies. The dimensions of multimedia communication include:

1. Text: The use of written words is an important aspect of multimedia communication. It can be used for conveying information, instructions, or opinions.
2. Audio: This dimension involves the use of sound in multimedia communication. It can be used to enhance the user experience or to convey information.
3. Video: Video is another important dimension of multimedia communication. It can be used to demonstrate a product or service or to tell a story.
4. Graphics: Graphics is a visual element of multimedia communication. It can be used to enhance text or to convey information such as graphs, charts, and images.
5. Animation: Animations are dynamic visual elements used in multimedia communication. It can be used to show motion, illustrate complex concepts, or add visual interest.
6. Interactivity: This dimension of multimedia communication refers to the ability of users to interact with the content. It can be used to create engaging experiences or to provide feedback.
7. Delivery platforms: The platforms through which multimedia is delivered also play an important role. These can range from traditional delivery methods like television and print media to newer platforms like social media and mobile devices.
8. Integration: Integration refers to the various media types being combined into a cohesive whole. For example, a multimedia presentation may use text, graphics and video together to communicate effectively.

1.3. Multimedia Communication Applications

Multimedia communication applications refer to software programs that allow users to exchange multimedia messages over the internet, including text, video, audio, and images. These applications typically support real-time communication, enabling users to engage in live conversations with others in remote locations. Some popular multimedia communication applications include:

1. Skype - a popular video and voice calling software that allows users to make free calls to other Skype users and low-cost calls to landlines and mobile phones.
2. WhatsApp - a messaging app that supports text, voice, and video calls between users, as well as group chat features and file sharing.
3. Zoom - a video conferencing app that enables users to host or attend virtual meetings, webinars, and online classes.
4. Google Meet - a web-based video conferencing app that enables users to host or attend virtual meetings and webinars, as well as share screens and collaborate on documents.
5. Viber - a messaging app that supports voice and video calls, messaging, and file sharing between users.
6. Microsoft Teams - a collaboration platform that allows users to communicate, share files, and collaborate on projects, as well as host virtual meetings and webinars.
7. Facetime - Apple's video and voice calling app that allows users to make calls between Apple devices.

1.4. Streaming and Downloading

Streaming refers to playing media content (such as video and audio) over the internet via a third-party service, without downloading the media file to the device. This means that the content is being "streamed" in real-time and does not take up any space on your device's hard drive.

Downloading, on the other hand, refers to the process of transferring a media file from a remote server to your device's hard drive, where it can be stored and accessed later. Once downloaded, the content can be played at any time without an internet connection.

The main difference between streaming and downloading is that streaming requires a steady internet connection to play the content in real-time, while downloading allows you to access the content offline. Additionally, streaming is often more convenient for accessing quick or on-the-go content, while downloading is better for long-term storage and access to content that you may want to watch or listen to repeatedly.

1.5. Streaming on Demand

Streaming Media on Demand refers to a technology that allows users to access media content, such as audio or video, on their devices in real time over the internet. Users can select and play the desired media content instantly, without the need for downloading or storing the media file on their device. This technology is widely used for watching movies or TV shows, listening to music, and playing video games. Popular examples of on-demand streaming services include Netflix, Hulu, Amazon Prime Video, and Spotify.

1.6. Live Broadcast

Live broadcast refers to the real-time transmission of audio and visual content, such as news, events, sports, music, and other performances, over a communication network to a large audience. The broadcast may be transmitted through radio, television, or the Internet and is typically delivered simultaneously to viewers or listeners. Live broadcasts allow people to experience events as they happen, without delay, and in the comfort of their own homes or on the go.

1.7. Real-time Communication

Real-time communication (RTC) refers to any form of communication that enables people to exchange information instantly or with minimal delay. This includes various forms of digital communication, like video conferencing, chat, And instant messaging applications.

RTC is possible thanks to advancements in digital technologies and the internet. For example, video conferencing software allows people to communicate with each other in real-time regardless of their physical location.

RTC has become particularly important with the rise of remote work and distributed teams. By enabling people to communicate and collaborate in real-time, RTC tools have become essential for businesses and organizations to remain productive and competitive.

Some examples of popular RTC tools include Zoom, Skype, Slack, WhatsApp, and Google Meet.

1.8. Multimedia Networking Architecture

Multimedia networks protocols architecture refers to the set of protocols and standards used to transmit and manage multimedia data over a network. These protocols are designed to ensure efficient and reliable transfer of multimedia data such as audio, video, and images between devices over a network.

The architecture of multimedia networks protocols can be divided into several layers, each of which performs a specific set of functions. These layers are often referred to as the OSI (Open Systems Interconnection) model.

Here is a brief overview of the different layers in the multimedia networks protocols architecture:

1. Physical layer: This layer is responsible for transmitting the raw data over a physical medium, such as a cable or wireless connection. It defines the physical characteristics of the transmission medium, such as the voltage levels, frequency, and modulation scheme.

2. Data link layer: This layer is responsible for framing the data into packets and adding error correction codes to ensure reliable transmission. It also manages access to the physical medium and handles flow control.

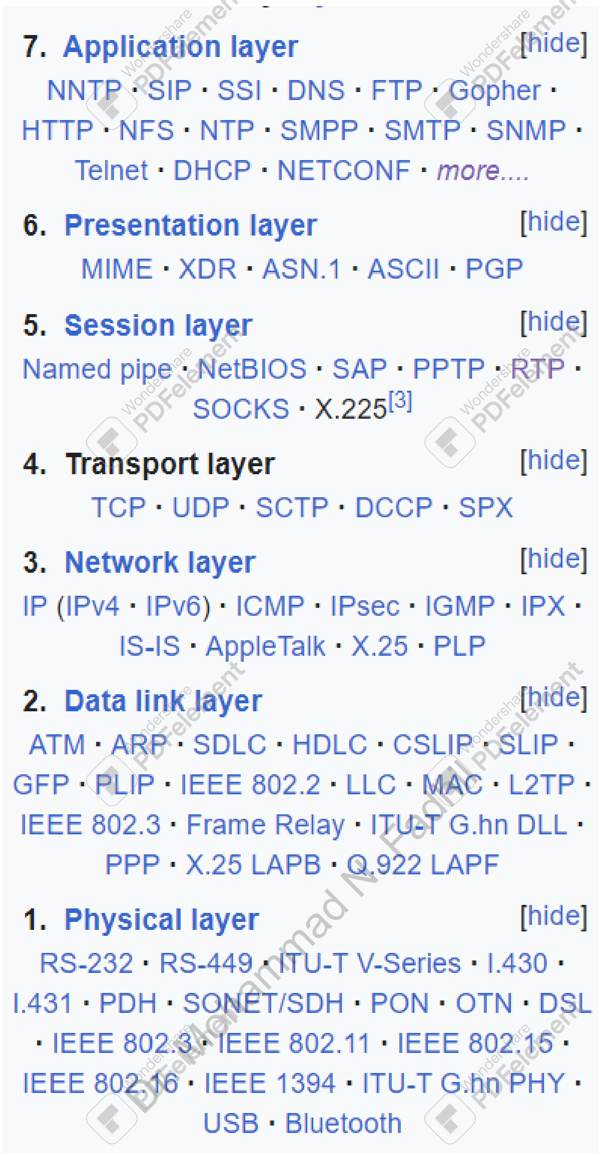
3. Network layer: This layer is responsible for routing the packets through the network to their destination. It manages the addressing and routing of packets and may also handle congestion control.

4. Transport layer: This layer is responsible for providing end-to-end data delivery, ensuring that the packets are delivered reliably and in order. It also provides error checking and recovery mechanisms.

5. Session layer: This layer is responsible for establishing, maintaining, and terminating communication sessions between devices. It manages the setup and synchronization of multimedia streams.

6. Presentation layer: This layer is responsible for encoding and decoding multimedia data, such as audio and video, into a format that can be understood by the application layer.

7. Application layer: This layer is responsible for providing the interface between the user and the network. It defines the protocols and applications that run on top of the lower layers and allows the user to interact with multimedia data.


7. Application layer [hide]
NNTP · SIP · SSI · DNS · FTP · Gopher · HTTP · NFS · NTP · SMPP · SMTP · SNMP · Telnet · DHCP · NETCONF · *more....*

6. Presentation layer [hide]
MIME · XDR · ASN.1 · ASCII · PGP

5. Session layer [hide]
Named pipe · NetBIOS · SAP · PPTP · RTP · SOCKS · X.225[3]

4. Transport layer [hide]
TCP · UDP · SCTP · DCCP · SPX

3. Network layer [hide]
IP (IPv4 · IPv6) · ICMP · IPsec · IGMP · IPX · IS-IS · AppleTalk · X.25 · PLP

2. Data link layer [hide]
ATM · ARP · SDLC · HDLC · CSLIP · SLIP · GFP · PLIP · IEEE 802.2 · LLC · MAC · L2TP · IEEE 802.3 · Frame Relay · ITU-T G.hn DLL · PPP · X.25 LAPB · Q.922 LAPF

1. Physical layer [hide]
RS-232 · RS-449 · ITU-T V-Series · I.430 · I.431 · PDH · SONET/SDH · PON · OTN · DSL · IEEE 802.3 · IEEE 802.11 · IEEE 802.15 · IEEE 802.16 · IEEE 1394 · ITU-T G.hn PHY · USB · Bluetooth

1.9. Multimedia Compression

Multimedia compression refers to the process of reducing the size of multimedia data to make it easier and more efficient to transmit and store. Multimedia data, such as images, audio, and video, often have a large file size that can take up significant storage space and require high bandwidth for transmission.

There are two types of multimedia compression: lossless and lossy compression.

1. Lossless compression: In lossless compression, the compressed data can be decompressed to an exact replica of the original data without any loss of information. Lossless compression is useful for multimedia data that requires

exact reproduction, such as medical images, text documents, and software files. Examples of lossless compression algorithms include ZIP, RAR, and PNG.

2. Lossy compression: In lossy compression, some data is lost during the compression process, resulting in a lower quality version of the original data. Lossy compression is useful for multimedia data that can tolerate some loss of information, such as images and video. Examples of lossy compression algorithms include JPEG, MPEG, and MP3.

Here are some common multimedia compression techniques:

1. Image Compression: Image compression techniques include JPEG, PNG, and GIF. These techniques use different algorithms to compress images, such as reducing the number of colors, using spatial redundancies, or exploiting human visual perception.
2. Audio Compression: Audio compression techniques include MP3, AAC, and WMA. These techniques use algorithms that remove sounds that are not audible to the human ear, reduce the bit rate, and compress the audio data.
3. Video Compression: Video compression techniques include MPEG, H.264, and VP9. These techniques use algorithms that remove redundant information, predict motion, and compress video frames.

Overall, multimedia compression is essential for efficient transmission and storage of multimedia data. The choice of compression technique depends on the type of multimedia data, the quality requirements, and the available bandwidth and storage capacity.

1.10. Multimedia Streaming

Multimedia streaming refers to the continuous transmission of multimedia content over the internet. This content may include video, audio, images, and text. Streaming technology allows users to consume this content in real-time, without having to download it to their devices first.

When you stream multimedia content, your device requests data from a server in small chunks, called packets. As each packet arrives, it is buffered and played back to the user, creating the impression of a continuous stream. The server sends data packets continuously until the end of the content is reached.

Multimedia streaming can be delivered through a variety of technologies, such as HTTP Live Streaming (HLS), Dynamic Adaptive Streaming over HTTP (DASH), Real-Time Messaging Protocol (RTMP), and MPEG-DASH. These technologies allow multimedia content to be optimized for different devices and network conditions, ensuring a smooth streaming experience for users.

Streaming services like Netflix, YouTube, and Spotify have become increasingly popular in recent years, and the technology behind multimedia streaming has played a significant role in their success. By enabling users to access and consume content seamlessly, streaming has revolutionized the way we consume multimedia content.

1.11. Interactive and non-Interactive Multimedia

Interactive multimedia refers to any digital content that can be interacted with by the user. This includes various forms of media, such as videos, audio, images, text, and animations, that are designed to respond to user input. Interactive multimedia is often used in education, entertainment, and marketing, and can be found in a variety of formats, including websites, apps, games, and simulations.

Examples of interactive multimedia include:

- Interactive videos: Videos that allow the user to make choices or interact with the content in some way.
- Games: Games that are designed to be interactive and respond to user input.

- Simulations: Programs that allow users to simulate real-world scenarios and interact with them.
- Interactive presentations: Presentations that allow the user to interact with the content, such as clicking on links, playing videos, or exploring interactive images.

Non-interactive multimedia, on the other hand, refers to digital content that is designed to be consumed without any user interaction. This includes videos, images, audio recordings, and other forms of media that are meant to be watched, listened to, or viewed. Non-interactive multimedia is often used in advertising, education, and entertainment, and can be found in a variety of formats, including television, movies, and digital media.

Examples of non-interactive multimedia include:

- Movies: Films that are designed to be watched without any user interaction.
- Television shows: Shows that are designed to be watched without any user interaction.
- Music recordings: Audio recordings that are designed to be listened to without any user interaction.
- Digital art: Artwork that is designed to be viewed without any user interaction.

1.12. Multimedia Distribution Model

Multimedia distribution models refer to the various methods through which multimedia content (such as images, audio, video, and text) is distributed to consumers or users. There are several multimedia distribution models, including:

1. Unicast: Unicast is a one-to-one communication model in which multimedia content is delivered from one sender to one receiver. This is similar to a telephone conversation, where the sender speaks to one receiver, and only that receiver hears the message. In the case of multimedia distribution, the sender may be a

streaming server or a content delivery network, and the receiver may be a personal computer or a mobile device. Unicast is commonly used for on-demand streaming of multimedia content such as movies or music, where the content is delivered to a specific user on request.

2. Multicast: Multicast is a one-to-many communication model in which multimedia content is delivered from one sender to multiple receivers simultaneously. This is similar to a radio broadcast, where one station transmits the same signal to multiple listeners. In the case of multimedia distribution, the sender may be a streaming server or a content delivery network, and the receivers may be multiple personal computers or mobile devices. Multicast is commonly used for live streaming of multimedia content such as sports events or concerts, where many users want to receive the same content at the same time.
3. Broadcast: Broadcast is a one-to-all communication model in which multimedia content is delivered from one sender to all receivers on the network. This is similar to a television broadcast, where one station transmits the same signal to all television sets in the coverage area. In the case of multimedia distribution, the sender may be a broadcasting station, and the receivers may be multiple personal computers or mobile devices within the coverage area. Broadcast is commonly used for distributing multimedia content such as news and emergency alerts to a wide audience.

Each of these communication models has its own advantages and disadvantages, and the choice of a distribution model depends on various factors such as the type of multimedia content, target audience, and network bandwidth.

1.13. Transmission Control Protocol (TCP) for Multimedia Networks

Multimedia over TCP refers to the use of the Transmission Control Protocol (TCP) for the transmission of multimedia data over computer networks. Multimedia refers to data that includes different forms of media such as audio, video, images, and text.

TCP is a reliable, connection-oriented protocol that ensures the delivery of data packets without any loss or corruption. This makes it an ideal choice for the transmission of multimedia data, which requires a high degree of reliability.

When multimedia data is transmitted over TCP, it is divided into small packets, which are then transmitted over the network. Each packet contains a sequence number, which helps to ensure that the packets are delivered in the correct order. If any packets are lost or corrupted during transmission, TCP uses retransmission and error correction mechanisms to ensure that the missing or damaged packets are retransmitted.

One of the advantages of using TCP for multimedia transmission is that it ensures that the data is transmitted at a consistent rate, which is important for maintaining the quality of the multimedia content. TCP also provides congestion control, which helps to prevent network congestion and ensures that the available network resources are used efficiently.

However, there are also some disadvantages to using TCP for multimedia transmission. One of the main disadvantages is that TCP is a relatively slow protocol, which can result in delays in the transmission of multimedia data. Additionally, TCP is not well-suited for real-time multimedia applications, such as video conferencing or live streaming, which require low latency and fast data transmission. In these cases, alternative protocols such as User Datagram Protocol (UDP) or Real-time Transport Protocol (RTP) are often used instead.

1.14. The Significant of User Datagram Protocol (UDP) in Multimedia Networks

UDP (User Datagram Protocol) is a communication protocol that operates at the transport layer of the Internet Protocol (IP) suite. It is a connectionless protocol that provides a low-overhead data delivery service without the reliability guarantees of TCP (Transmission Control Protocol). In multimedia applications, UDP is often used to transmit real-time data, such as video and audio, because it provides fast and efficient transmission with minimal delay.

The significance of UDP in multimedia lies in its ability to deliver real-time data streams with low latency and minimal loss. This is particularly important in applications where real-time delivery is critical, such as video conferencing, online gaming, and live streaming. With UDP, data is transmitted in small packets that are sent as quickly as possible, without waiting for confirmation from the receiver. This means that data can be sent more quickly and with lower overhead than with TCP.

Another advantage of UDP in multimedia applications is that it allows for multicast and broadcast transmission. Multicast allows a single data stream to be transmitted to multiple recipients simultaneously, while broadcast transmission sends data to all devices on a network. This is particularly useful for live events or other situations where multiple users need to receive the same data simultaneously.

However, the downside of UDP is that it does not provide any error correction or retransmission mechanisms, so data packets can be lost or corrupted during transmission. This can result in a degradation of the quality of the multimedia stream. To mitigate this, multimedia applications often use techniques such as forward error correction (FEC) or retransmission protocols to ensure reliable delivery.

In summary, UDP is significant in multimedia applications because it provides a fast and efficient delivery mechanism for real-time data streams with low latency, and allows for multicast and broadcast transmission. However, it lacks the reliability guarantees of TCP and can result in data loss or corruption if not handled properly.

2. Real-Time Transport Protocol (RTP)

RTP stands for "Real-time Transport Protocol," and it is a protocol used for transmitting audio and video data over networks, particularly the Internet.

RTP is designed to provide end-to-end delivery of real-time data, such as audio and video streams, between applications. It does this by breaking up the data into small, manageable packets, each with its own header containing information about the data payload and its sequence. RTP uses UDP as the transport protocol because it provides a low-latency, best-effort delivery service, which is suitable for real-time media transmission.. In the context of networking and communication, low-latency refers to the time it takes for data to travel from one point to another in a network. This is typically measured in milliseconds (ms), and lower latency means that data can be transmitted more quickly, which is important for real-time applications such as video conferencing, online gaming, and financial trading.

One of the key features of RTP is its ability to provide timing and synchronization information to ensure that real-time data is played back in the correct sequence and at the appropriate speed. RTP also includes mechanisms for error detection and correction, as well as support for encryption and authentication.

RTP is often used in conjunction with other protocols, such as the Session Initiation Protocol (SIP) or the H.323 protocol suite, to establish and control the transmission of real-time data between applications. It is commonly used for applications such as voice and video conferencing, online gaming, and live streaming of multimedia content.

RTP is designed for end-to-end, real-time transfer of streaming media. The protocol provides facilities for jitter compensation and detection of packet loss and out-of-order delivery, which are common especially during UDP transmissions on an IP network. RTP allows data transfer to multiple destinations through IP multicast. RTP is regarded as the primary standard for audio/video transport in IP networks and is used with an associated profile and payload format. The design of RTP is based on the architectural principle known as application-layer framing where protocol functions are implemented in the application as opposed to the operating system's protocol stack.

2.1. RTP Profiles and payload formats

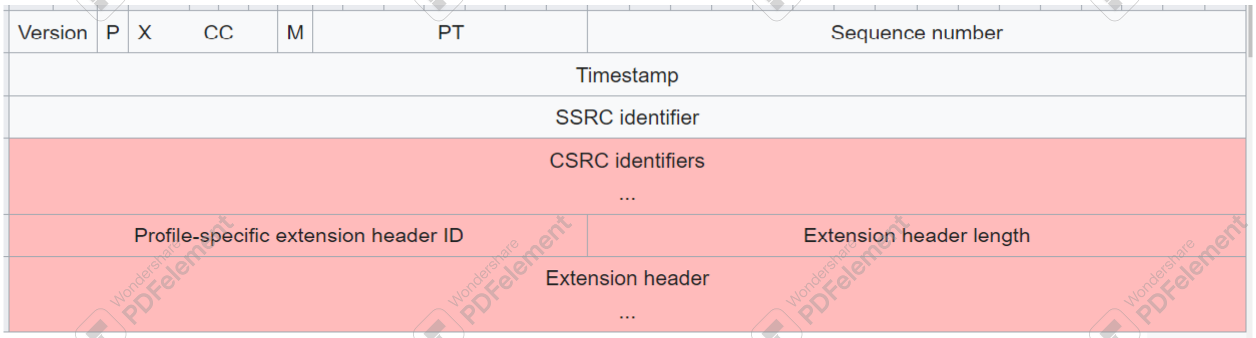
RTP is designed to carry a multitude of multimedia formats, which permits the development of new formats without revising the RTP standard. To this end, the information required by a specific application of the protocol is not included in the generic RTP header. For each class of application (e.g., audio, video), RTP defines a profile and associated payload formats. Every instantiation of RTP in a particular application requires a profile and payload format specifications.

The profile defines the codecs used to encode the payload data and their mapping to payload format codes in the protocol field Payload Type (PT) of the RTP header. Each profile is accompanied by several payload format specifications, each of which describes the transport of particular encoded data. Examples of audio payload formats are G.711, G.723, G.726, G.729, GSM, QCELP, MP3, and DTMF, and examples of video payloads are H.261, H.263, H.264, H.265 and MPEG-1/MPEG-2.

2.2. RTP Packet header

RTP packets are created at the application layer and handed to the transport layer for delivery. Each unit of RTP media data created by an application begins with the RTP packet header.

The RTP header has a minimum size of 12 bytes. After the header, optional header extensions may be present. This is followed by the RTP payload, the format of which is determined by the particular class of application. The fields in the header are as follows:



Version	P	X	CC	M	PT	Sequence number
Timestamp						
SSRC identifier						
CSRC identifiers ...						
Profile-specific extension header ID						Extension header length
Extension header ...						

Version: (2 bits) Indicates the version of the protocol.

P (Padding): (1 bit) Used to indicate if there are extra padding bytes at the end of the RTP packet. Padding may be used to fill up a block of certain size, for example as required by an encryption algorithm. The last byte of the padding contains the number of padding bytes that were added (including itself).

X (Extension): (1 bit) Indicates presence of an extension header between the header and payload data. The extension header is application or profile specific.

CC (CSRC count): (4 bits) Contains the number of CSRC identifiers (defined below) that follow the SSRC (also defined below).

M (Marker): (1 bit) Signaling used at the application level in a profile-specific manner. If it is set, it means that the current data has some special relevance for the application.

PT (Payload type): (7 bits) Indicates the format of the payload and thus determines its interpretation by the application. Values are profile specific and may be dynamically assigned.

Sequence number: (16 bits) The sequence number is incremented for each RTP data packet sent and is to be used by the receiver to detect packet loss[3] and to accommodate out-of-order delivery. The initial value of the sequence number should be randomized to make known-plaintext attacks on Secure Real-time Transport Protocol more difficult.

Timestamp: (32 bits) Used by the receiver to play back the received samples at appropriate time and interval. When several media streams are present, the timestamps may be independent in each stream.[b] The granularity of the timing is application specific. For example, an audio application that samples data once every 125 µs (8 kHz, a common sample rate in digital telephony) would use that value as its clock resolution. Video streams typically use a 90 kHz clock. The clock granularity is one of the details that is specified in the RTP profile for an application.

SSRC: (32 bits) Synchronization source identifier uniquely identifies the source of a stream. The synchronization sources within the same RTP session will be unique.

CSRC: (32 bits each, the number of entries is indicated by the CSRC count field) Contributing source IDs enumerate contributing sources to a stream which has been generated from multiple sources.

Header extension: (optional, presence indicated by Extension field) The first 32-bit word contains a profile-specific identifier (16 bits) and a length specifier (16 bits) that indicates the length of the extension in 32-bit units, excluding the 32 bits of the extension header.

2.3. RTP Application design

A functional multimedia application requires other protocols and standards used in conjunction with RTP. Protocols such as SIP, Jingle, RTSP, H.225 and H.245 are used for session initiation, control and termination. Other standards, such as H.264, MPEG and H.263, are used for encoding the payload data as specified by the applicable RTP profile.

An RTP sender captures the multimedia data, then encodes, frames and transmits it as RTP packets with appropriate timestamps and increasing timestamps and sequence numbers. The sender sets the payload type field in accordance with connection negotiation and the RTP profile in use. The RTP receiver detects missing packets and may reorder packets. It decodes the media data in the packets according to the payload type and presents the stream to its user.

3. Real-time Control Protocol (RTCP)

Real-time control protocol (RTCP) is a protocol used in conjunction with the real-time transport protocol (RTP) to monitor the quality of service (QoS) for media streams, such as audio and video, that are being transmitted over a network. The RTCP is responsible for providing feedback on the transmission statistics of RTP packets, including packet loss, delay, and jitter.

1. RTCP is a control protocol that works in conjunction with RTP, which is responsible for transmitting the actual multimedia data.

2. The primary function of RTCP is to provide feedback about the quality of the RTP stream, including statistics such as packet loss, delay, and jitter.
3. RTCP is typically sent periodically by the receiving endpoint (receiver) to the transmitting endpoint (sender).
4. The RTCP packets contain information about the receiver's buffer size, number of packets lost, and other statistics that the sender can use to adjust its transmission rate.
5. The RTCP packets can also be used to convey information about the session, such as the identity of the participants and the session's start and end times.
6. RTCP is designed to be lightweight and not interfere with the real-time nature of the multimedia stream.
7. RTCP operates on a separate port from RTP. The port used for RTCP is typically the next highest odd port number after the port used for RTP. For example, if RTP is using port 5000, then RTCP would use port 5001.

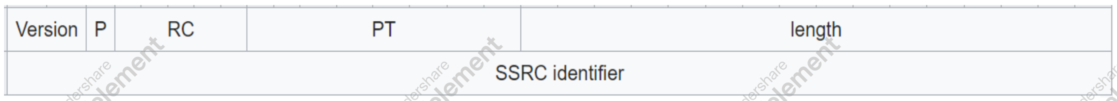
3.1. RTCP Protocol functions

RTCP provides basic functions expected to be implemented in all RTP sessions:

- The primary function of RTCP is to gather statistics on quality aspects of the media distribution during a session and transmit this data to the session media source and other session participants. Such information may be used by the source for adaptive media encoding (codec) and detection of transmission faults. If the session is carried over a multicast network, this permits non-intrusive session quality monitoring.
- RTCP provides canonical end-point identifiers (CNAME) to all session participants. Although a source identifier (SSRC) of an RTP stream is expected to be unique, the instantaneous binding of source identifiers to end-points may change during a session. The CNAME establishes unique identification of end-points across an application instance (multiple use of media tools) and for third-party monitoring.

- Provisioning of session control functions. RTCP is a convenient means to reach all session participants, whereas RTP itself is not. RTP is only transmitted by a media source.

3.2. RTCP Packet header



- Version: (2 bits) Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. The version defined by this specification is two (2).
- P (Padding): (1 bits) Indicates if there are extra padding bytes at the end of the RTP packet. Padding may be used to fill up a block of certain size, for example as required by an encryption algorithm. The last byte of the padding contains the number of padding bytes that were added (including itself).
- RC (Reception report count): (5 bits) The number of reception report blocks contained in this packet. A value of zero is valid.
- PT (Packet type): (8 bits) Contains a constant to identify RTCP packet type.
- Length: (16 bits) Indicates the length of this RTCP packet (including the header itself) in 32-bit units minus one.
- SSRC: (32 bits) Synchronization source identifier uniquely identifies the source of a stream.

3.3. RTCP Message types

RTCP distinguishes several types of packets: sender report, receiver report, source description, and goodbye. In addition, the protocol is extensible and allows application specific RTCP packets.

- **Sender report (SR)**

  The sender report is sent periodically by the active senders in a conference to report transmission and reception statistics for all RTP packets sent during the interval. The sender report includes two distinct timestamps, an absolute timestamp, represented using the timestamp format of the Network Time Protocol (NTP) and an RTP timestamp that corresponds to the same time as the NTP timestamp. The absolute timestamp allows the receiver to synchronize RTP messages. It is particularly important when both audio and video are transmitted simultaneously because audio and video streams use independent relative timestamps.

- **Receiver report (RR)**

The receiver report is for passive participants, those that do not send RTP packets. The report informs the sender and other receivers about the quality of service.

- **Source description (SDES)**

The Source Description message is used to send the CNAME item to session participants. It may also be used to provide additional information such as the name, e-mail address, telephone number, and address of the owner or controller of the source.

- **Goodbye (BYE)**

A source sends a BYE message to shut down a stream. It allows an endpoint to announce that it is leaving the conference. Although other sources can detect the absence of a source, this message is a direct announcement. It is also useful to a media mixer.

- **Application-specific message (APP)**

The application-specific message provides a mechanism to design application-specific extensions to the RTCP protocol.

4. Real Time Streaming Protocol (RTSP)

The Real Time Streaming Protocol (RTSP) is an application-level network protocol designed for multiplexing and packetizing multimedia transport streams (such as interactive media, video and audio) over a suitable transport protocol. RTSP is used in entertainment and communications systems to control streaming media servers. The protocol is used for establishing and controlling media sessions between endpoints. Clients of media servers issue commands such as play, record and pause, to facilitate real-time control of the media streaming from the server to a client (video on demand) or from a client to the server (voice recording).

5. Quality of Service (QoS)

Quality of service (QoS) is the description or measurement of the overall performance of a service, such as a telephony or computer network, or a cloud computing service, particularly the performance seen by the users of the network. To quantitatively measure quality of service, several related aspects of the network service are often considered, such as packet loss, bit rate, throughput, transmission delay, availability, jitter, etc.

In the field of computer networking and other packet-switched telecommunication networks, quality of service refers to traffic prioritization and resource reservation control mechanisms rather than the achieved service quality. Quality of service is the ability to provide different priorities to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow.

Quality of service is particularly important for the transport of traffic with special requirements. In particular, developers have introduced Voice over IP technology to allow computer networks to become as useful as telephone networks for audio conversations, as well as supporting new applications with even stricter network performance requirements.

Quality of service comprises requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, crosstalk, echo, interrupts, frequency response, loudness levels, and so on. A subset of telephony QoS is grade of service (GoS) requirements, which comprises aspects of a connection relating to capacity and coverage of a network, for example guaranteed maximum blocking probability and outage probability.

In the field of computer networking and other packet-switched telecommunication networks, teletraffic engineering refers to traffic prioritization and resource reservation control mechanisms rather than the achieved service quality. Quality of service is the ability to provide different priorities to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. For example, a required bit rate, delay, delay variation, packet loss or bit error rates may be guaranteed. Quality of service is important for real-time streaming multimedia applications such as voice over IP, multiplayer online games and IPTV, since these often require fixed bit rate and are delay sensitive. Quality of service is especially important in networks where the capacity is a limited resource, for example in cellular data communication.

A network or protocol that supports QoS may agree on a traffic contract with the application software and reserve capacity in the network nodes, for example during a session establishment phase. During the session it may monitor the achieved level of performance, for example the data rate and delay, and dynamically control scheduling priorities in the network nodes. It may release the reserved capacity during a tear down phase.

5.1. Qualities of Traffic

In packet-switched networks, quality of service is affected by various factors, which can be divided into human and technical factors. Human factors include: stability of service quality, availability of service, waiting times and user information. Technical factors include: reliability, scalability, effectiveness, maintainability and network congestion.

Many things can happen to packets as they travel from origin to destination, resulting in the following problems as seen from the point of view of the sender and receiver:

- **Goodput**

  Due to varying load from disparate users sharing the same network resources, the maximum throughput that can be provided to a certain data stream may be too low for real-time multimedia services.

- **Packet loss**

  The network may fail to deliver (drop) some packets due to network congestion. The receiving application may ask for this information to be retransmitted, possibly resulting in congestive collapse or unacceptable delays in the overall transmission.

- **Errors**

  Sometimes packets are corrupted due to bit errors caused by noise and interference, especially in wireless communications and long copper wires. The receiver has to detect this, and, just as if the packet was dropped, may ask for this information to be retransmitted.

- **Latency**

  It might take a long time for each packet to reach its destination because it gets held up in long queues, or it takes a less direct route to avoid congestion. In some cases, excessive latency can render an application such as VoIP or online gaming unusable.

- **Packet delay variation**

  Packets from the source will reach the destination with different delays. A packet's delay varies with its position in the queues of the routers along the path between source and destination, and this position can vary unpredictably. Delay variation can be absorbed at the receiver, but in so doing increases the overall latency for the stream.

- **Out-of-order delivery**

  When a collection of related packets is routed through a network, different packets may take different routes, each resulting in a different delay. The result is that the packets arrive in a different order than they were sent. This problem requires special additional protocols for rearranging out-of-order packets. The reordering process requires additional buffering at the receiver, and, as with packet delay variation, increases the overall latency for the stream.

5.2. QoS Protocols

Several QoS mechanisms and schemes exist for IP networking.

- The type of service (ToS) field in the IPv4 header.
- Differentiated services (DiffServ)
- Integrated services (IntServ)
- Resource Reservation Protocol (RSVP)
- RSVP-TE

QoS capabilities are available in the following network technologies.

- Multiprotocol Label Switching (MPLS) provides eight QoS classes
- Frame Relay
- X.25
- Some DSL modems
- ATM
- Ethernet supporting IEEE 802.1Q with Audio Video Bridging and Time-Sensitive Networking
- Wi-Fi supporting IEEE 802.11e

5. Session Initiation Protocol

Session Initiation Protocol (SIP) forms the backbone of modern real-time communication networks. Over the years, SIP has been enhanced a great deal to include several use cases that make it a very robust multipurpose communication protocol. The following sections provide a brief overview of SIP.

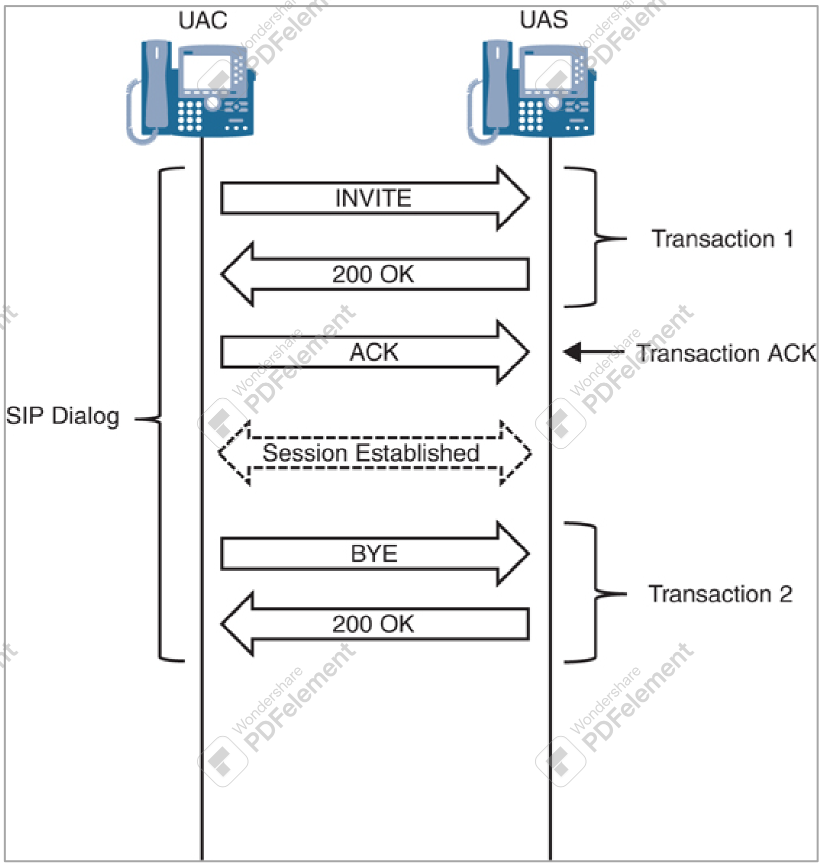
The SIP communications protocol is used for session setup, modification, and teardown. It is an application layer protocol that incorporates many elements of Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP). SIP is modular in design and can work in concert with many other protocols that are required to set up and support communication sessions, including the following:

- Real-Time Transport Protocol (RTP)
- Session Description Protocol (SDP)
- Resource Reservation Protocol (RSVP)

5.1. SIP Operation

SIP works on the request/response framework and mirrors a model similar to HTTP, where there is a client/server exchange. A node that generates the request is called a user agent client (UAC), and a node that processes the request and sends out at least one response is called a user agent server (UAS). The concepts of a SIP transaction and a SIP dialog characterize the interaction between the UACs and UASs. A SIP transaction consists of a single request and all responses to that request, which may include zero or more provisional responses (1XX) and one or more final responses (2XX, 3XX, 4XX, 5XX, 6XX). A SIP dialog is a peer-to-peer relationship between user agents that exists for some time.

A single SIP dialog can include multiple SIP transactions. A transaction consists of a single request and the response(s) to that request. Figure below shows a few of these messages and an example of the SIP dialog with multiple transactions. The first transaction in this figure, known as the INVITE transaction, forms the SIP three-way handshake observed in many SIP dialogs.



Sample SIP Dialog with Two Transactions

The following events occur in Transaction 1 in this example:

- The UAC sends an INVITE message to the UAS in an attempt to establish a session.
- The UAS sends a 200 OK response, which accepts the INVITE message for the session.
- The UAC needs to acknowledge the 200 OK message for the INVITE transaction, which is done via an ACK request message. (Note that this message is only observed during the INVITE transaction.)

At this point, the session is established, and it continues until one participant decides to end the session. When that occurs, a second transaction is created, consisting of the following events:

- Session teardown occurs, with a BYE message sent by one participant (in this case, the originating UAC).
- The UAS accepts the BYE for session teardown and replies by sending a 200 OK.

NOTE

1XX messages, which are optional, are omitted from Figure 2-1. Similarly, the type of session (for example, audio, video) being established is omitted to keep the example simple.

SIP commonly uses TCP or UDP as the transport protocol. For devices such as call agents, voice gateways, SIP proxies, and session border controllers (SBCs) that typically handle several SIP sessions simultaneously, the transport layer protocol is usually UDP. Establishing and maintaining a connection involves significantly more overhead with TCP than with UDP. However, when SIP sessions need to traverse communication links that are prone to errors, such as packet drops, it is better to use TCP as the transport layer protocol. Port number 5060 is typically used for SIP over UDP or TCP.

SIP messages exchanged between UACs and UASs carry a lot of information that could be misused if it fell into the hands of an attacker. For example, a SIP INVITE carries information that could reveal details of the network topology, the nature of the device originating or servicing the request, and details of the media stream(s), such as the IP addresses and port numbers. This is especially problematic when communication sessions span open networks. To prevent such attacks, it is possible to secure SIP signaling by using Transport Layer Security (TLS). The port number used for SIP over TLS is 5061.
Resources on a SIP network are identified by a uniform resource identifier (URI), which takes the following generic format:

```
sip:username:password@host:port
```

If the port is not specified, it defaults to 5060. For secure SIP transmission over TLS, an s may be added to the end of sip to make it sips:

```
sips:username:password@host:port
```

Note that the sips URI is not required when using TLS, and many implementations use SIP over TLS with the sip URI. As with a non-secure sip URI, if the port is not specified, a default port is used. In this case, however, the default port is 5061 instead of 5060.

SIP devices are referred to as user agents and can be devices such as IP phones, call servers, fax servers, gateways, and SBCs. The originator of a SIP request is called a UAC, and a device that processes the request is called a UAS. SIP includes several functional components, and interactions between user agents in real-world scenarios are in most cases more complex than generic client/server transactions. In order to understand the core tenets of SIP operation, you need to understand the following functional components of SIP:

- **SIP proxy:** A SIP proxy is a device that is capable of performing call routing, authentication, authorization, address resolution, loop detection, and load balancing. A SIP proxy can be stateless or stateful; the fundamental difference between the two is whether they are aware of SIP transactions. A SIP transaction consists of a single request and all responses to that request, which may include zero or more provisional responses (1XX) and one or more final responses.
  A stateful proxy becomes aware of the state of a SIP transaction by creating a server transaction, a client transaction, and a response context. By being transaction aware, the proxy is capable of forking requests, retransmitting requests, and generating messages by itself. *For example,* a stateful SIP proxy can generate a SIP CANCEL message to all entities still processing a forked request after a final response has already been received.

Stateless proxies, on the other hand, do not maintain transaction state; they transparently forward requests from the client to the server, and they send responses in the reverse direction. Once a request or a response is forwarded to the intended recipient, all details or transaction context of the message is purged. Consequently, stateless proxies cannot fork requests, retransmit requests, or generate messages on their own.

Proxies do not manipulate SIP message headers such as To, From, Call-ID, and so on. They do, however, include a Via header and a Record-Route header, and they decrease the Max-Forwards header value by one.

- **Redirect server:** A redirect server is a server that provides location services to user agents or proxies by replying to requests with the location or route to the host that ultimately services the request. This is desirable in situations where there is a need to build highly scalable servers that do not participate in a SIP transaction but simply help the proxy or user agent reach the host by sending a single message. Redirect servers reply to requests with a 3XX response in which the Contact header contains the URI of the location to the host.
- **Registrar server:** A registrar server is a server that accepts registration requests from user agents and creates a mapping between their address of record (AOR)—the public identifier of the user agent—and the user agent's location.
- **Location server:** A location server contains a mapping between a user agent's AOR and location; it does not need to be by a separate physical server and can be physically and logically colocated with the registrar server.
- B2BUA: Back-to-back user agents (B2BUAs) are devices that have both UAC and UAS functionality and are capable of forwarding requests and processing them. SBCs, such as CUBE, are examples of B2BUAs.

5.2. SIP Methods

SIP messages are transmitted in plaintext. SIP messages can be requests or responses to requests. Table below lists SIP requests and describes the purpose of each one.

SIP Request	Description
INVITE	A caller sends out this message to request another entity to establish a SIP session.
ACK	This indicates that the client has received a final response to an INVITE request.
BYE	This is used by the UAC to indicate to the server that it wishes to terminate the established SIP session. (Note that this request can be issued by the caller or the callee.)
CANCEL	This is used to cancel a pending request and can be sent only if the server has not replied with a final response.
REGISTER	A client uses this message to register the address listed in the To header field with a SIP registrar server.
PRACK	This provisional acknowledgment is used to ensure that provisional responses are received reliably.
NOTIFY	This notifies the subscriber of the occurrence of an event.
PUBLISH	This publishes an event to the server.
UPDATE	This modifies the state of the session without changing the state of the dialog.

Every SIP transaction begins with a request from a UAC to a UAS. The UAS begins processing the request as soon as it is received. The result of this processing depends on the nature of the request, the formatting of the request, the state of the server at the time the request was being serviced, and the general configuration and policies local to the server. In the case of devices such as SIP proxies, B2BUAs, and voice gateways, the result of processing a request could depend on downstream devices.

SIP servers are always required to respond with the results of request processing. SIP responses use the following formatting convention:

- The SIP version number (2.0 is the current SIP version number)
- A three-digit status code (for example, 404)
- A textual description (for example, Not Found)

The three-digit status code is an integer that communicates the outcome of request processing and is used for machine interpretation. The textual description, on the other hand, is for human observers and is useful in call failure debugging and call record interpretation. The first digit of the status code indicates the SIP response class; there are six classes in all. (See Table below).

Class of Response	**Meaning**	**Type of Response**
1XX	Informational: The request has been received and is being processed.	Provisional
2XX	Success: The request has been received, understood, and accepted.	Final
3XX	Redirection: Further action needs to be taken to complete the request. For example, the UAC needs to contact another server to process the request.	Final
4XX	Client error: The request contains bad syntax (such as malformed headers) or could not be fulfilled at the server (for example, if the server could not find the number referenced in the requested URI).	Final
5XX	Server error: A server failed to fulfill a valid request.	Final
6XX	Global failure: The request could not be fulfilled at any server.	Final

6. H.323

H.323 is a recommendation from the ITU Telecommunication Standardization Sector (ITU-T) that defines the protocols to provide audio-visual communication sessions on any packet network. The H.323 standard addresses call signaling and control, multimedia transport and control, and bandwidth control for point-to-point and multi-point conferences.

H.323 is widely implemented by voice and video conferencing equipment manufacturers, is used within various Internet real-time applications such as NetMeeting and is widely deployed worldwide by service providers and enterprises for both voice and video services over IP networks.

H.323 is a VoIP call control protocol that allows for the establishment, maintenance, and teardown of multimedia sessions across H.323 endpoints. H.323 is a suite of specifications that controls the transmission of voice, video, and data over IP networks. The following are some of the H.323 specifications:

- H.225: H.225 handles call setup and teardown between H.323 endpoints and is also responsible for peering with H.323 gatekeepers via the Registration Admission Status (RAS) protocol.
- H.245: H.245 acts as a peer protocol to H.225 and is used to negotiate the characteristics of the media session, such as media format, the method of DTMF relay, the media type (audio, video, fax, and so on), and the IP address/port pair for media.
- H.450: H.450 controls supplementary services between H.323 entities. These supplementary services include call hold, call transfer, call park, and call pickup.

6.1. H.323 Components

The H.323 protocols outlined in the previous section are used in the communications between H.323 components or devices. The following are the most common H.323 devices:

- H.323 gateways: H.323 gateways are endpoints that are capable of interworking between a packet network and a traditional Plain Old Telephone Service (POTS) network (analog or digital). Since these H.323 endpoints can implement their own call routing logic, they are considered to be “intelligent” and, as such, operate in a peer-to-peer mode. H.323 gateways are capable of registering to a gatekeeper and interworking calls with a gatekeeper by using the RAS protocol.

- H.323 gatekeepers: H.323 gatekeepers function as devices that provide lookup services. They indicate via signaling to which endpoint or endpoints a particular called number belongs. Gatekeepers also provide functionality such as Call Admission Control and security. Endpoints register to the gatekeeper by using the RAS protocol.

- H.323 terminals: Any H.323 device that is capable of setting up a two-way, real-time media session is an H.323 terminal. H.323 terminals include voice gateways, H.323 trunks, video conferencing stations, and IP phones. H.323 terminals use H.225 for session setup, progress, and teardown. They also use H.245 to define characteristics of the media session such as the media format, the method of DTMF, and the media type.

- Multipoint control units: These H.323 devices handle multiparty conferences, and each device is composed of a multipoint controller (MC) and multipoint processor (MP). The MC is responsible for H.245 exchanges, and the MP is responsible for the switching and manipulation of media.

6.2. H.323 Call Flow

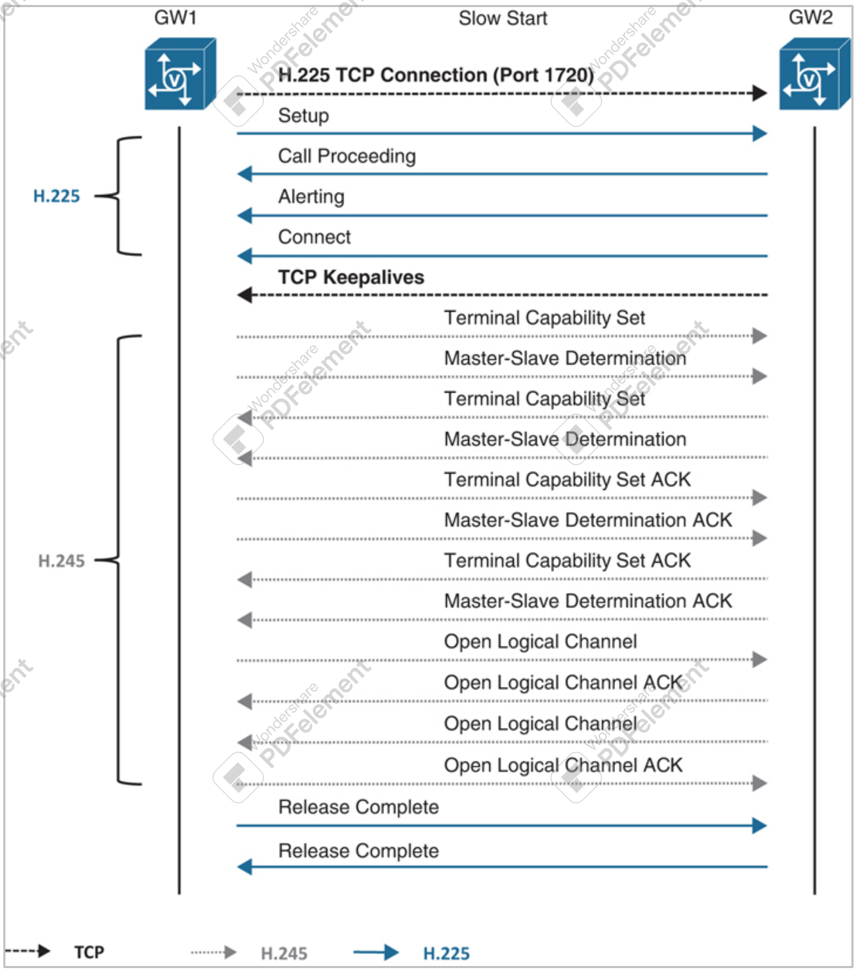
An H.323 call basically involves the following:

- A TCP socket must be established on port 1720 to initiate H.225 signaling with another H.323 peer. This assumes that there is no gatekeeper in the call flow. As defined in the previous section, gatekeepers assist in endpoint discovery and call admission.
- For an H.323 call, the H.225 exchange is responsible for call setup and termination, whereas the H.245 exchange is responsible for establishment of the media channels and their properties. In most cases, the establishment of two

independent TCP connections is required: one for the H.225 exchange and the other for the H.245 exchange. To effectively bind the two, the TCP port number on which the answering terminal intends to establish an H.245 exchange is advertised in one of the H.225 messages. The port number can be advertised before the H.225 connect message is sent (for example, in an H.225 progress message) or when the H.225 connect message is sent.

- H.225 and H.245 exchanges can proceed on the same TCP connection, using a process called H.245 tunneling.
- Every H.245 message is unidirectional in the sense that it is used to specify the negotiation from the perspective of the sender of that H.245 message. For the successful establishment of a two-way real-time session, both H.323 terminals must exchange H.245 messages.

Figure below depicts a basic H.323 slow start call between two H.323 terminals. The calling terminal first initiates a TCP connection to the called terminal, using destination port 1720. Once this connection is established, H.225 messages are exchanged between the two terminals to set up the call. In order to negotiate parameters that define call characteristics such as the media types (for example, audio, video, fax), media formats, and DTMF types, an H.245 exchange has to ensue between the terminals.



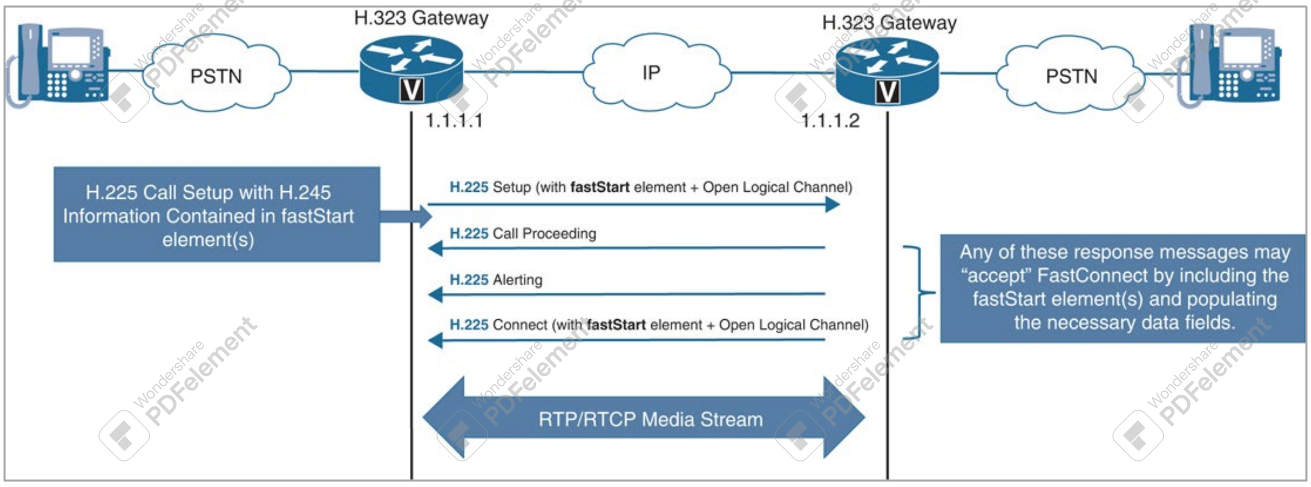
Basic H.323 Slow Start Call

In most cases, a separate TCP connection is established between the endpoints to negotiate an H.245 exchange; however, in some cases, as an optimization, H.245 messages are tunneled using the same TCP socket as H.225, using a procedure known as H.245 tunneling. When utilizing a separate TCP connection for H.245, the called terminal advertises the TCP port number over which it intends to establish an H.245 exchange. The ports used for the establishment of H.245 are ephemeral and are not dictated by the H.323 specification.

The H.245 exchange results in the establishment of the media channels required to transmit and receive real-time information. You should be aware that while Figure 2-7 highlights a slow start call, a variant to the slow start procedure, known as FastConnect, also exists and is depicted in Figure 2-8. As the name suggests, FastConnect is a quicker and more efficient mechanism to establish an H.323 call. In fact, FastConnect can establish an H.323 call with as few as two messages. This is possible because with FastConnect there is no need to open an H.245 socket, as long as all needed media can be negotiated via FastConnect.

> NOTE
>
> FastConnect is often improperly referred to as “fast start” or “fastStart,” after the name of the associated field/element in H.225 messages that is used to negotiate and establish FastConnect.

Figure 2-8 shows how an H.323 FastConnect call is set up. When transmitting a Call Setup message, the endpoint populates a field, known as the fastStart element, with H.245 messages. The called endpoint can accept FastConnect by selecting any fastStart element in the Call Setup message, populate the necessary data fields (as specified in H.323), and return a fastStart element in any H.225 message (for example, Call Proceeding, Alerting, Connect) to the caller. The called endpoint can also reject FastConnect and fall back to the traditional slow start procedures shown in Figure below by either explicitly indicating so (using a flag), initiating any H.245 communications, or providing an H.245 address for the purposes of initiating H.245 communications.


H.323 Gateway
PSTN
1.1.1.1
IP
H.323 Gateway
1.1.1.2
PSTN
H.225 Call Setup with H.245 Information Contained in fastStart element(s)
H.225 Setup (with fastStart element + Open Logical Channel)
H.225 Call Proceeding
H.225 Alerting
H.225 Connect (with fastStart element + Open Logical Channel)
Any of these response messages may “accept” FastConnect by including the fastStart element(s) and populating the necessary data fields.
RTP/RTCP Media Stream

H.323 FastConnect Call Setup

H.323 Gatekeeper

An H.323 Gatekeeper serves the purpose of Call Admission Control and translation services from IDs (commonly a phone number) to IP addresses in an H.323 telephony network. Gatekeepers can be combined with a gateway function to proxy H.323 calls and are sometimes referred to as Session Border Controllers. A gatekeeper can also deny access or limit the number of simultaneous connections to prevent network congestion.

H.323 endpoints are not required to register with a gatekeeper to be able to place point to point calls, but they are essential for any serious H.323 network to control call prefix routing and link capacities among other functions.

A typical H.323 Gatekeeper call flow for a successful call may look like:-

```
   | |                           |  |
Endpoint A                    Endpoint B
  1234                          1123
```

1. Endpoint A dials 1123 from the system.
2. Endpoint A sends ARQ (Admission Request) to the Gatekeeper.
3. Gatekeeper returns ACF (Admission Confirmation) with IP address of endpoint B.
4. Endpoint A sends Q.931 call setup messages to endpoint B.
5. Endpoint B sends the Gatekeeper an ARQ, asking if it can answer call.
6. Gatekeeper returns an ACF with IP address of endpoint A.
7. Endpoint B answers and sends Q.931 call setup messages to endpoint A.
8. Information Request Response(IRR) sent to Gatekeeper from both endpoints.
9. Either endpoint disconnects the call by sending a DRQ (Disconnect Request) to the Gatekeeper.
10. Gatekeeper sends a DCF (Disconnect Confirmation) to both endpoints.

Multipoint Control Unit: The Multipoint Control Unit (MCU) is an endpoint on the network which provides the capability for three or more terminals and Gateways to participate in a multipoint conference.

Endpoint: An H.323 terminal, Gateway, or MCU. An endpoint can call and be called. It generates and/or terminates information streams

Gatekeeper: The Gatekeeper (GK) is an H.323 entity on the network that provides address translation and controls access to the network for H.323 terminals, Gateways and MCUs. The Gatekeeper may also provide other services to the terminals, Gateways and MCUs such as bandwidth management and locating Gateways.