MODULE 5: MULTIMEDIA NETWORKING

MODULE 5: MULTIMEDIA NETWORKING

5.1 Multimedia Networking Applications

• A multimedia network application can be defined as any application that employs audio or video.

5.1.1 Properties of Video

1) High Bit Rate

• Video distributed over the Internet use
  → 100 kbps for low-quality video conferencing.
  → 3 Mbps for streaming high-definition (HD) movies.

• The higher the bit-rate,
  → better the image quality and
  → better the overall user viewing experience.

2) Video Compression

• A video can be compressed, thereby trading off video-quality with bit-rate.
• A video is a sequence of images, displayed at a constant rate.
For example: 24 or 30 images per second.
• An uncompressed digital image consists of an array of pixels.
• Each pixel is encoded into a number of bits to represent luminance and color.
• There are two types of redundancy in video:

1) Spatial Redundancy
➢ An image that consists of mostly white space has a high degree of redundancy.
➢ These images can be efficiently compressed without sacrificing image quality.

2) Temporal Redundancy
➢ Temporal redundancy reflects repetition from image to subsequent image.
➢ For exanple:
If image & subsequent image are same, re-encoding of subsequent image can be avoided.

5.1.2 Properties of Audio
• PCM (Pulse Code Modulation) is a technique used to change an analog signal to digital data (digitization).
• PCM consists of 1) Encoder at the sender and 2) Decoder at the receiver.

5.1.2.1 PCM Encoder
• Digital audio has lower bandwidth requirements than video.
• Consider how analog audio is converted to a digital-signal:
• The analog audio-signal is sampled at some fixed rate. This operation is referred to as sampling.
• For example: 8000 samples per second.
• The value of each sample is an arbitrary real number.
• Each sample is then rounded to one of a finite number of values. This process is called quantization.
• The number of such finite values is called as quantization-values.
• The number of quantization-values is typically a power of 2. For ex: 256(2^8) quantization-values.
• Each of the quantization-values is represented by a fixed number of bits.
• For example:
  If there are 256(2^8) quantization-values, then each value is represented by 8 bits.
• Bit representations of all values are then concatenated to form digital representation of the signal. This process is called encoding.
• For example:
  If an analog-signal is sampled at 8000 samples per second & each sample is represented by 8 bits, then the digital-signal will have a rate of 64000 bits per second (8000*8=64000).

5.1.2.2 PCM Decoder
• For playback through audio speakers, the digital-signal can be converted back to an analog-signal. This process is called decoding.
• However, the decoded analog-signal is only an approximation of the original signal.
• The sound quality may be noticeably degraded.
• The decoded signal can better approximate the original analog-signal by increasing
  i) sampling rate and
  ii) number of quantization-values,
• Thus, there is a trade-off between
  → quality of the decoded signal and
  → bit-rate & storage requirements of the digital-signal.

5.1.3 Types of Multimedia Network Applications
• Three broad categories of multimedia applications:
1) Streaming stored audio/video
2) Conversational voice/video-over-IP and
3) Streaming live audio/video.

5.1.3.1 Streaming Stored Audio & Video
• The underlying medium is prerecorded video. For example: a movie.
• These prerecorded videos are placed on servers.
• The users send requests to the servers to view the videos on-demand.
• Nowadays, many Internet companies provide streaming video. For example: YouTube.
• Three key distinguishing features of streaming stored video:

1) Streaming
• The client begins video playout within few seconds after it begins receiving the video from the server.
• At the same time,
i) The client will be playing out from one location in the video.
ii) The client will be receiving later parts of the video from the server.
• This technique avoids having to download the entire video-file before playout begins.

2) Interactivity
• The media is prerecorded, so the user may pause, reposition or fast-forward through video-content.
• The response time should be less than a few seconds.

3) Continuous Playout
• Once playout of the video begins, it should proceed according to the original timing of the recording.
• The data must be received from the server in time for its playout at the client.
Otherwise, users experience video-frame skipping (or freezing).

5.1.3.2 Conversational Voice- and Video-over-IP
• Real-time conversational voice over the Internet is often referred to as Internet telephony.
• It is also commonly called Voice-over-IP (VoIP).
• Conversational video includes the video of the participants as well as their voices.
• Most of today's voice applications allow users to create conferences with three or more participants.
• Nowadays, many Internet companies provide voice application. For example: Skype & Google Talk.
• Two parameters are particularly important for voice applications:
1) Timing considerations and
2) Tolerance of data loss
• Timing considerations are important because voice applications are highly delay-sensitive.
• Loss-tolerant means
Occasional loss only causes occasional glitches in audio playback & these losses can be partially/fully hidden.

5.1.3.3 Streaming Live Audio & Video
• These applications are similar to broadcast radio, except that transmission takes place over Internet.
• These applications allow a user to receive a live radio transmitted from any corner of the world.
• For example: live cricket commentary.
• Today, thousands of radio stations around the world are broadcasting content over the Internet.
• Live broadcast applications have many users who receive the same audio program at the same time.
• The network must provide an average throughput that is larger than the video consumption rate.

5.2 Streaming Stored Video

- Prerecorded videos are placed on servers.
- Users send requests to these servers to view the videos on-demand.
- The media is prerecorded, so the user may pause, reposition or fast-forward through video-content.
- Three categories of applications:
    1) UDP streaming
    2) HTTP streaming and
    3) Adaptive HTTP streaming.
- A main characteristic of video-streaming is the extensive use of client-side buffering.
- Two advantages of client-side buffering:
    1) Client-side buffering can mitigate effects of varying end-to-end delays
    2) This can mitigate effects of varying amounts of available bandwidth b/w server & client.

5.2.1 UDP Streaming

- The server transmits video at a rate that matches the client's video consumption rate.
- The server transmits the video-chunks over UDP at a steady rate.
- UDP does not employ a congestion-control mechanism.
- Therefore, the server can push packets into the network at the video consumption rate.
- Typically, UDP streaming uses a small client-side buffer. (RTP → Real-Time Transport Protocol).
- Using RTP, the server encapsulates the video-chunks within transport packets.
- The client & server also maintain a control-connection over which the client sends commands (such as pause, resume and reposition).
- The RTSP (Real-Time Streaming Protocol) is a popular open protocol for a control-connection.
- Disadvantages:

    1) Unreliability

    ➢ UDP streaming can fail to provide continuous playout '.' of varying amt of available bandwidth

    2) Costly & Complex

    ➢ A media control server (RTSP) is required

    → to process client-to-server interactivity requests and

    → to track client-state for each ongoing client-session.

    ➢ This increases the overall cost and complexity of deploying a large-scale application.

    3) Firewall Problem

    ➢ Many firewalls are configured to block UDP traffic.

    ➢ This prevents the users behind the firewalls from receiving the video.

5.2.2 HTTP Streaming

- The video is stored in an HTTP server as an ordinary file with a specific URL.
- Here is how it works:
  1) When a user wants to see the video, the client
     → establishes a TCP connection with the server and
     → issues an HTTP GET request for that URL.
  2) Then, the server responds with the video file, within an HTTP response message.
  3) On client side, the bytes are collected in a client application buffer.
  4) Once no. of bytes in this buffer exceeds a specific threshold, the client begins playback.
- Advantages:
  1) Not Costly & Complex
  ➢ Streaming over HTTP avoids the need for a media control server (RTSP).
  ➢ This reduces the cost of deploying a large-scale application.
  2) No Firewall Problem
  ➢ The use of HTTP over TCP also allows the video to traverse firewalls and NATs more easily.
  3) Prefetching Video
  ➢ The client downloads the video at a rate higher than the consumption rate.
  ➢ Thus, prefetching video-frames that are to be consumed in the future.
  ➢ This prefetched video is stored in the client application buffer
- Nowadays, most video-streaming applications use HTTP streaming. For example: YouTube

5.2.2.1 Client Application Buffer & TCP Buffers

- Figure 5.1 illustrates the interaction between client and server for HTTP streaming.
- On the server side,
  1) The bytes of the video file are sent into the server's socket.
  2) Then, the bytes are placed in the TCP send buffer before.
  3) Finally, the bytes are transmitted into the Internet.
- On the client side,
  1) The application (media-player) reads bytes from the TCP receive-buffer (thro client-socket)
  2) Then, the application places the bytes into the client-buffer.
  3) At the same time, the application periodically
     → grabs video-frames from the client-buffer
     → decompresses the frames and
     → displays the frames on the user's screen.

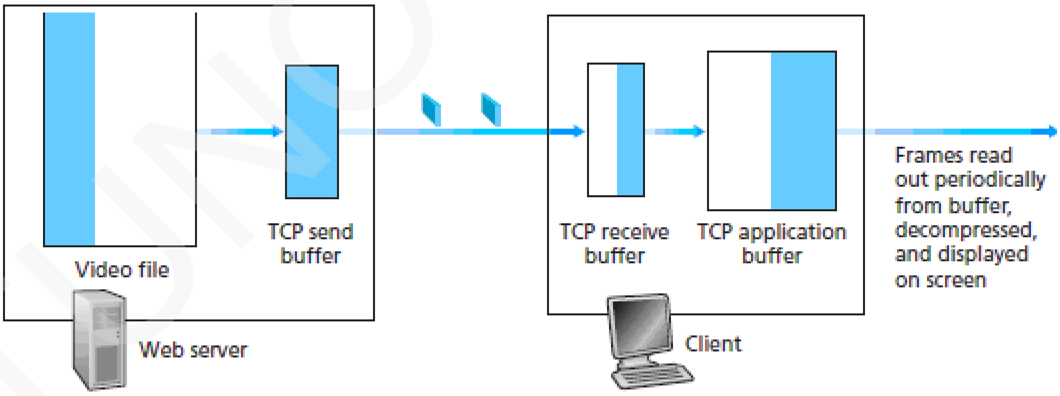

Figure 5.1: Streaming stored video over HTTP/TCP

5.2.2.2 Early Termination & Repositioning the Video

- HTTP streaming systems make use of the byte-range header in the HTTP GET request message.
- Byte-range header specifies the range of bytes the client currently wants to retrieve from the video.
- This is particularly useful when the user wants to reposition to a future point in the video.
- When the user repositions to a new position, the client sends a new HTTP request.
- When server receives new HTTP request, the server sends the requested-bytes.
- Disadvantage:
  When a user repositions to a future point in the video, some prefetched-but-not-yet-viewed data will go unwatched. This results in a waste of bandwidth and server-resources.

5.2.3 Adaptive Streaming & DASH

• Problem with HTTP streaming:

All clients receive the same encoding of video, despite the large variations in the amount of bandwidth available to different clients.

Solution: Use DASH (Dynamic Adaptive Streaming over HTTP).

5.2.3.1 DASH

• The video is encoded into several different versions.
• Each version has a different bit-rate and a different quality level.
• Two main tasks:
  1) The client dynamically requests video-chunks from the different versions: low & high.
     i) When the available bandwidth is high, the client selects chunks from a high-rate version.
        For ex: Fiber connections can receive a high-quality version.
     ii) When the available bandwidth is low, the client naturally selects from a low-rate version.
        For ex: 3G connections can receive a low-quality version.
  2) The client adapts to the available bandwidth if end-to-end bandwidth changes during session.
     ➢ This feature is particularly important for mobile-users.
     ➢ The mobile-users see their bandwidth fluctuate as they move with respect to base-stations.
• HTTP server stores following files:
  1) Each video version with a different URL.
  2) Manifest file provides a URL for each version along with its bit-rate.
• Here is how it works:
  1) First, the client requests the manifest file and learns about the various versions.
  2) Then, the client selects one chunk at a time by specifying
     → URL and
     → byte range in an HTTP GET request message.
  3) While downloading chunks, the client
     → measures the received bandwidth and
     → runs a rate determination-algorithm.
        i) If measured-bandwidth is high, client will choose chunk from high-rate version.
        ii) If measured-bandwidth is low, client will choose chunk from low-rate version
  4) Therefore, DASH allows the client to freely switch among different quality-levels.
• Advantages:
  1) DASH can achieve continuous playout at the best possible quality level w/o frame freezing.
  2) Server-side scalability is improved: Because
     → the client maintains the intelligence to determine which chunk to send next.
  3) Client can use HTTP byte-range request to precisely control the amount of prefetched video.

5.2.4 CDN

5.2.4.1 Motivation for CDN

• The streaming video service can be provided is as follows:

1) Build a single massive data-center.
2) Store all videos in the data-center and
3) Stream the videos directly from the data-center to clients worldwide.

• Three major problems with the above approach:

1) More Delay

➢ If links provides a throughput lesser than consumption-rate, the end-to-end throughput will also be below the consumption-rate.

➢ This results in freezing delays for the user.

2) Network Bandwidth is wasted

➢ A popular video may be sent many times over the same links.

3) Single Point of Failure:

➢ If the data-center goes down, it cannot distribute any video streams.

Problem Solution: Use CDN (Content Distribution Network).

5.2.4.2 CDN Types

• A CDN

→ manages servers in multiple geographically distributed locations
→ stores copies of the videos in its servers, and
→ attempts to direct each user-request to a CDN that provides the best user experience.

• The CDN may be a private CDN or a third-party CDN.

1) Private CDN

➢ A private CDN is owned by the content provider itself.

➢ For example:

Google's CDN distributes YouTube videos

2) Third Party CDN

➢A third-party CDN distributes content on behalf of multiple content providers CDNs.

➢ Two approaches for server placement:

i) Enter Deep

¤ The first approach is to enter deep into the access networks of ISPs.
¤ Server-clusters are deployed in access networks of ISPs all over the world.
¤ The goal is to get close to end users.
¤ This improves delay/throughput by decreasing no. of links b/w end user & CDN cluster

ii) Bring Home

¤ The second approach is to bring the ISPs home.
¤ Large clusters are built at a smaller number of key locations.
¤ These clusters are connected using a private high-speed network.
¤ Typically, clusters are placed at a location that is near the PoPs of many tier-1 ISPs.
For example: within a few miles of both Airtel and BSNL PoPs in a major city.
¤ Advantage:
Lower maintenance and management overhead.
¤ Disadvantage:
Higher delay and lower throughput to end users.

5.2.4.3 CDN Operation

- When a browser wants to retrieve a specific video, the CDN intercepts the request.
- Then, the CDN
    1) determines a suitable server-cluster for the client and
    2) redirects the client's request to the desired server.
- Most CDNs take advantage of DNS to intercept and redirect requests.
- CDN operation is illustrated in Figure 5.2.

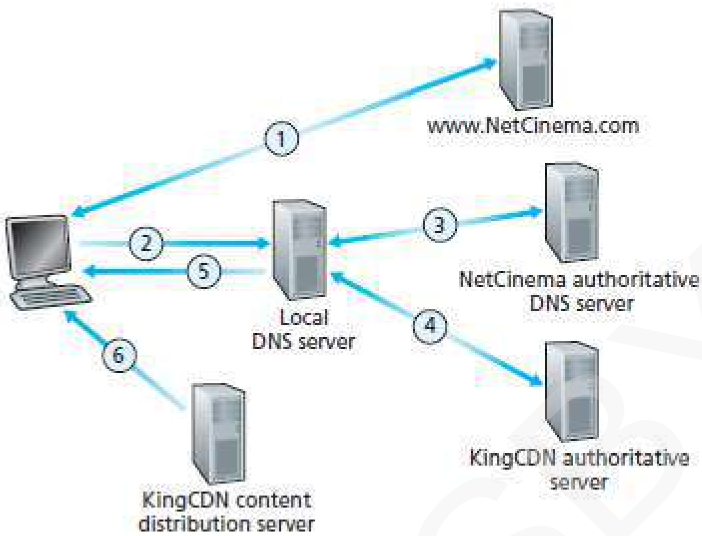

Figure 5.2: DNS redirects a user's request to a CDN server

- Suppose a content provider "NetCinema" employs the CDN company "KingCDN" to distribute videos.
- Let URL = http://video.netcinema.com/6Y7B23V
- Six events occur as shown in Figure 5.2:
    1) The user visits the Web page at NetCinema.
    2) The user clicks on the following link:
    http://video.netcinema.com/6Y7B23V,
    ➢ Then, the user's host sends a DNS query for "video.netcinema.com".
    3) The user's local-DNS-server (LDNS) forwards the DNS-query to an authoritative-DNS-server "NetCinema".
    ➢ The server "NetCinema" returns to the LDNS a hostname in the KingCDN's domain.
    ➢ For example: "a1105.kingcdn.com".
    4) The user's LDNS then sends a second query, now for "a1105.kingcdn.com".
    ➢ Eventually, KingCDN's DNS system returns the IP addresses of a "KingCDN" server to LDNS.
    5) The LDNS forwards the IP address of the "KingCDN" server to the user's host.
    6) Finally, the client
        → establishes a TCP connection with the server
        → issues an HTTP GET request for the video.

5.2.4.4 Cluster Selection Strategies

- Cluster-selection strategy is used for dynamically directing clients to a server-cluster within the CDN.
- The CDN learns the IP address of the client's LDNS server via the client's DNS lookup.
- After learning this IP address, the CDN selects an appropriate cluster based on this IP address.
- Three approaches for cluster-selection:

1) Geographically Closest

- ➢ The client is assigned to the cluster that is geographically closest.
- ➢ Using geo-location databases, each LDNS IP address is mapped to a geographic location.
- ➢ When a DNS request is received from LDNS, the CDN chooses geographically closest-cluster.
- ➢ Advantage:
  This solution can work reasonably well for a large fraction of the clients.
- ➢ Disadvantages: The solution may perform poorly. This is because
  1) Geographically closest-cluster may not be the closest-cluster along the path.
  2) The LDNs location may be far from the client's location.

2) Based on Current Traffic Conditions

- ➢ The best cluster can be determined for a client based on the current traffic-conditions.
- ➢ CDNs perform real-time measurements of delay/loss performance b/w their clusters & clients.
- ➢ In a CDN, each cluster periodically sends probes to all of the LDNSs around the world.
- ➢ Disadvantage:
  Many LDNSs are configured to not respond to the probes.

3) IP Anycast

- ➢ The idea behind IP anycast:
  In Internet, the routers must route the packets to the closest-cluster, as determined by BGP.
- ➢ IP anycast is illustrated in Figure 5.3.

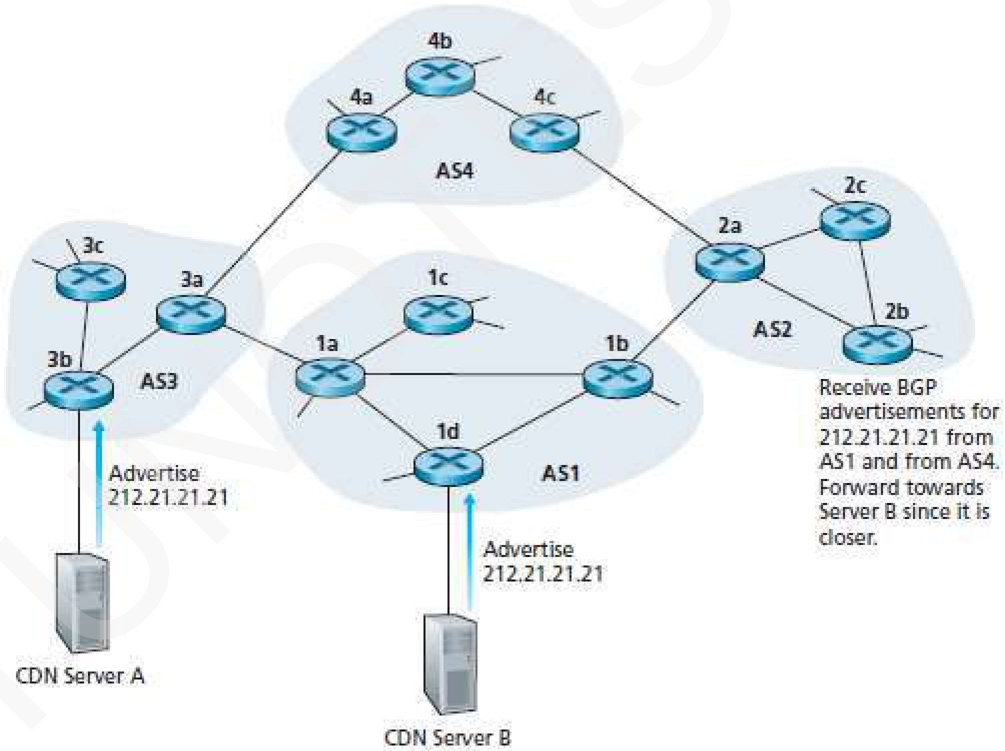

Figure 5.3: Using IP anycast to route clients to closest CDN cluster

- ➢ Here is how it works:
  1) During the IP-anycast configuration stage, the CDN company
     → assigns the same IP address to each clusters and
     → uses BGP to advertise the IP address from different cluster locations.
  2) The BGP router treats the multiple route advertisements as different paths to the same physical location.
  3) Then, the BGP router picks the "best" route to the IP address.

5.3 Voice-over-IP
• Real-time voice over the Internet is often referred to as Internet telephony.
• It is also commonly called Voice-over-IP (VoIP).

5.3.1 Limitations of the Best-Effort IP Service
• The Internet's network-layer protocol IP provides best-effort service.
• The IP makes best effort to move each datagram from source to destination.
• But IP does not guarantee deliver of the packet to the destination.
• Three main challenges to the design of real-time applications:
    1) Packet-loss
    2) Packet delay and
    3) Packet jitter.

5.3.1.1 Packet Loss
• By default, most existing VoIP applications run over UDP.
• The UDP segment is encapsulated in an IP datagram.
• The datagram passes through router buffers in the path from sender to receiver
• Problem:
    ➢ There is possibility that one or more buffers are full.
    ➢ In this case, the arriving IP datagram may be discarded.
• Possible solution:
    ➢ Loss can be eliminated by sending the packets over TCP rather than over UDP.
    ➢ However, retransmissions are unacceptable for real-time applications '.' they increase delay.
    ➢ Packet-loss results in a reduction of sender's transmission-rate, leading to buffer starvation.

5.3.1.2 End-to-End Delay
• End-to-end delay is the sum of following delays:
    1) Transmission, processing, and queuing delays in routers.
    2) Propagation delays in links and
    3) Processing delays in end-systems.
• For VoIP application,
    → delays smaller than 150 msecs are not perceived by a human listener.
    → delays between 150 and 400 msecs can be acceptable but are not ideal and
    → delays exceeding 400 msecs can seriously hinder the interactivity in voice conversations.
• Typically, the receiving-side will discard any packets that are delayed more than a certain threshold.
• For example: more than 400 msecs.

5.3.1.3 Packet Jitter
• Jitter refers to varying queuing delays that a packet experiences in the network's routers.
• If the receiver
    → ignores the presence of jitter and
    → plays out audio-chunks,
  then the resulting audio-quality can easily become unintelligible.
• Jitter can often be removed by using sequence numbers, timestamps, and a playout delay

5.3.2 Removing Jitter at the Receiver for Audio

• For VoIP application, receiver must provide periodic playout of voice-chunks in presence of random jitter

• This is typically done by combining the following 2 mechanisms:

1) Prepending each Chunk with a Timestamp

➢ The sender attaches each chunk with the time at which the chunk was generated.

2) Delaying Playout of Chunks at the Receiver

➢ The playout delay of the received chunks must be long.

➢ So, the most of the packets are received before their scheduled playout times.

➢ This playout delay can either be

→ fixed throughout the duration of the session or

→ vary adaptively during the session-lifetime.

5.3.3 Recovering from Packet Loss

- Loss recovery schemes attempt to preserve acceptable audio-quality in the presence of packet-loss.
- Here, packet-loss is defined in a 2 broad sense:
  - i) A packet is lost if the packet never arrives at the receiver or
  - ii) A packet is lost if the packet arrives after its scheduled playout time.
- VoIP applications often use loss anticipation schemes.
- Here, we consider 2 types of loss anticipation schemes:
  1) Forward error correction (FEC) and
  2) Interleaving.
- We also consider an error concealment scheme.

5.3.3.1 FEC

- The basic idea of FEC: Redundant information is added to the original packet stream.
- The redundant information can be used to reconstruct approximations of some of the lost-packets.
- Two FEC mechanisms:

1) Block Coding

- ➢ A redundant encoded chunk is sent after every n chunks.
- ➢ The redundant chunk is obtained by exclusive OR-ing the n original chunks.
- ➢ If any one packet in a group is lost, the receiver can fully reconstruct the lost-packet.
- ➢ Disadvantages:
  1) If 2 or more packets in a group are lost, receiver cannot reconstruct the lost-packets.
  2) Increases the playout delay. This is because
     → receiver must wait to receive entire group of packets before it can begin playout.

2) Lower Resolution Redundant Information

- ➢ A lower-resolution audio-stream is sent as the redundant information.
- ➢ For example: The sender creates
  → nominal audio-stream and
  → corresponding low-resolution, low-bit-rate audio-stream.
- ➢ The low-bit-rate stream is referred to as the redundant-stream.
- ➢ As shown in Figure 5.4, the sender constructs the nth packet by
  → taking the nth chunk from the nominal stream and
  → appending the nth chunk to the (n−1)st chunk from the redundant-stream
- ➢ Advantage:
  Whenever there is packet-loss, receiver can hide the loss by playing out low-bit-rate chunk.

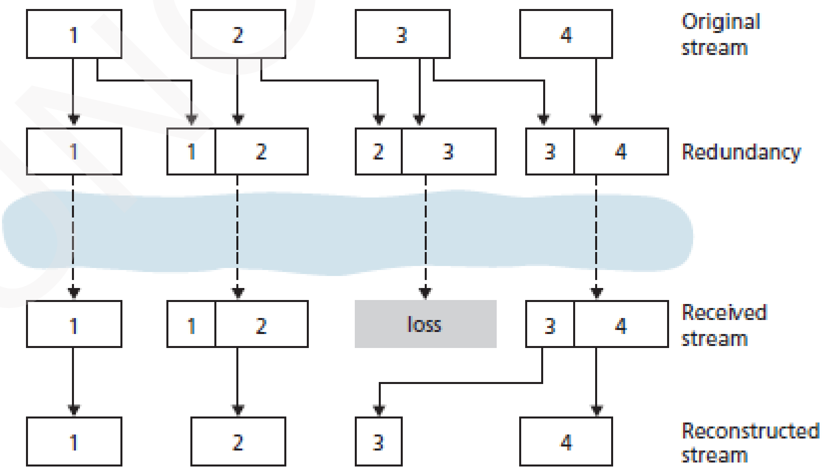

Figure 5.4: Piggybacking lower-quality redundant information

5.3.3.2 Interleaving

- A VoIP application can send interleaved audio.
- The sender resequences units of audio-data before transmission.
- Thus, originally adjacent units are separated by a certain distance in the transmitted-stream.
- Interleaving can mitigate the effect of packet-losses.
- Interleaving is illustrated in Figure 5.5.
- For example:

  If units are 5 msecs in length and chunks are 20 msecs (that is, four units per chunk), then

  → the first chunk contains the units 1, 5, 9, and 13

  → the second chunk contains the units 2, 6, 10 & 14 and so on.
- Advantages:

  1) Improves the perceived quality of an audio-stream.

  2) Low overhead.

  3) Does not increase the bandwidth requirements of a stream.
- Disadvantage:

  1) Increases latency. This limits use for VoIP applications.

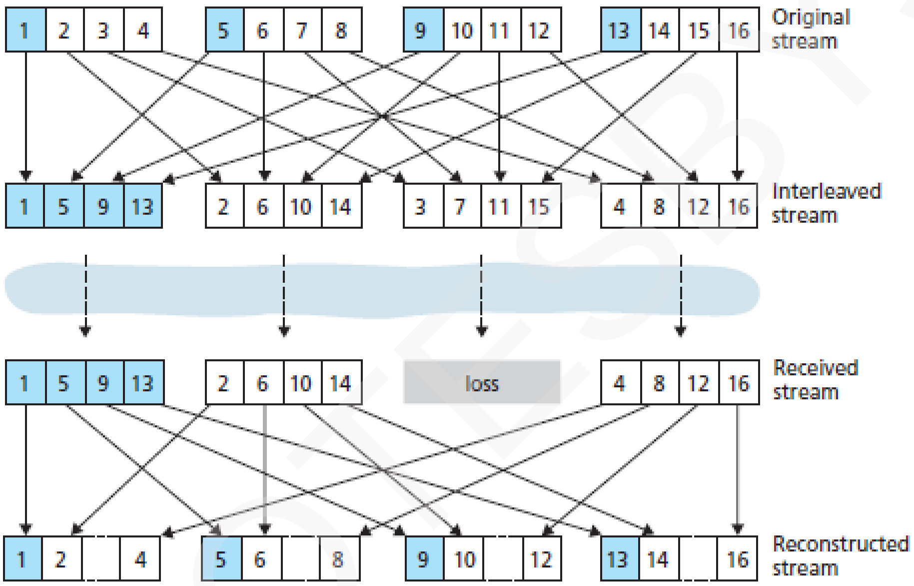

Figure 5.5: Sending interleaved audio

5.3.3.3 Error Concealment

- This scheme attempts to produce a replacement for a lost-packet that is similar to the original.
- This is possible since audio-signals exhibit large amounts of short-term self-similarity.
- Two receiver-based recovery techniques:

  1) Packet Repetition

  - This replaces lost-packets with copies of packets that arrived immediately before the loss.
  - Advantage:

    1) Low computational complexity.

  2) Interpolation

  - This uses audio before and after the loss to interpolate a suitable packet to cover the loss.
  - Advantage:

    1) Performs better than packet repetition.

5.4 Protocols for Real-Time Conversational Applications
- Real-time applications are very popular. For ex: VoIP and video conferencing.
- Two standards bodies are working for real-time applications: 1) IETF and 2) ITU
- Both standards (IETF & ITU) are enjoying widespread implementation in industry products.

5.4.1 RTP
- RTP can be used for transporting common formats such as
  → MP3 for sound and
  → MPEG for video
- It can also be used for transporting proprietary sound and video formats.
- Today, RTP enjoys widespread implementation in many products and research prototypes.
- It is also complementary to other important real-time interactive protocols, such as SIP.

5.4.1.1 RTP Basics
- RTP runs on top of UDP.
- The RTP packet is composed of i) RTP header & ii) audio chunk
- The header includes
  i) Type of audio encoding
  ii) Sequence number and
  iii) Timestamp.
- The application appends each chunk of the audio-data with an RTP header.
- Here is how it works:

1) At sender-side:
   i) A media chunk is encapsulated within an RTP packet.
   ii) Then, the packet is encapsulated within a UDP segment.
   iii) Finally, the UDP segment is handed over to IP.
2) At receiving-side:
   i) The RTP packet is extracted from the UDP segment.
   ii) Then, the media chunk is extracted from the RTP packet.
   iii) Finally, the media chunk is passed to the media-player for decoding and rendering

- If an application uses RTP then the application easily interoperates with other multimedia applications
- For example:
  If 2 different companies use RTP in their VoIP product, then users will be able to communicate.
- What RTP does not provide?
  i) It does not provide any mechanism to ensure timely delivery of data.
  ii) It does not provide quality-of-service (QoS) guarantees.
  iii) It does not guarantee delivery of packets.
  iv) It does not prevent out-of-order delivery of packets.
- RTP encapsulation is seen only at the end systems.
- Routers do not distinguish between
  i) IP datagrams that carry RTP packets and
  ii) IP datagrams that don't carry RTP packets.
- RTP allows each source to be assigned its own independent RTP stream of packets.
- For example:
  1) For a video conference between two participants, four RTP streams will be opened
     i) Two streams for transmitting the audio (one in each direction) and
     ii) Two streams for transmitting the video (again, one in each direction).
  2) Encoding technique MPEG bundles audio & video into a single stream.
  ➢ In this case, only one RTP stream is generated in each direction.
- RTP packets can also be sent over one-to-many and many-to-many multicast trees.

5.4.1.2 RTP Packet Header Fields

• Four header fields of RTP Packet (Figure 5.6):
  1) Payload type
  2) Sequence number
  3) Timestamp and
  4) Source identifier.

• Header fields are illustrated in Figure 5.6.

Payload type	Sequence number	Timestamp	Synchronization source identifier	Miscellaneous fields

Figure 5.6: RTP header fields

Table 5.1: Audio payload types supported by RTP

Payload-Type Number	Audio Format	Sampling Rate	Rate
0	PCM μ-law	8 kHz	64 kbps
1	1016	8 kHz	4.8 kbps
3	GSM	8 kHz	13 kbps
7	LPC	8 kHz	2.4 kbps
9	G.722	16 kHz	48–64 kbps
14	MPEG Audio	90 kHz	—
15	G.728	8 kHz	16 kbps

Table 5.2: Some video payload types supported by RTP

Payload-Type Number	Video Format
26	Motion JPEG
31	H.261
32	MPEG 1 video
33	MPEG 2 video

1) Payload Type

i) For an audio-stream, this field is used to indicate type of audio encoding that is being used.
  ➢ For example: PCM, delta modulation.
  ➢ Table 5.1 lists some of the audio payload types currently supported by RTP.

ii) For a video stream, this field is used to indicate the type of video encoding.
  ➢ For example: motion JPEG, MPEG.
  ➢ Table 5.2 lists some of the video payload types currently supported by RTP.

2) Sequence Number

• This field increments by one for each RTP packet sent.
• This field may be used by the receiver to detect packet loss and to restore packet sequence.

3) Timestamp

• This field reflects the sampling instant of the first byte in the RTP data packet.
• The receiver can use timestamps
  → to remove packet jitter in the network and
  → to provide synchronous playout at the receiver.
• The timestamp is derived from a sampling clock at the sender.

4) Source Identifier (SRC)

• This field identifies the source of the RTP stream.
• Typically, each stream in an RTP session has a distinct SRC.

5.4.2 SIP

- SIP (Session Initiation Protocol) is an open and lightweight protocol.
- Main functions of SIP:
  1) It provides mechanisms for establishing calls b/w a caller and a callee over an IP network.
  2) It allows the caller to notify the callee that it wants to start a call.
  3) It allows the participants to agree on media encodings.
  4) It also allows participants to end calls.
  5) It provides mechanisms for the caller to determine the current IP address of the callee.
  6) It provides mechanisms for call management, such as
     → adding new media streams during the call
     → changing the encoding during the call
     → inviting new participants during the call,
     → call transfer and
     → call holding.

5.4.2.1 Setting up a Call to a Known IP Address
• SIP call-establishment process is illustrated in Figure 5.7.

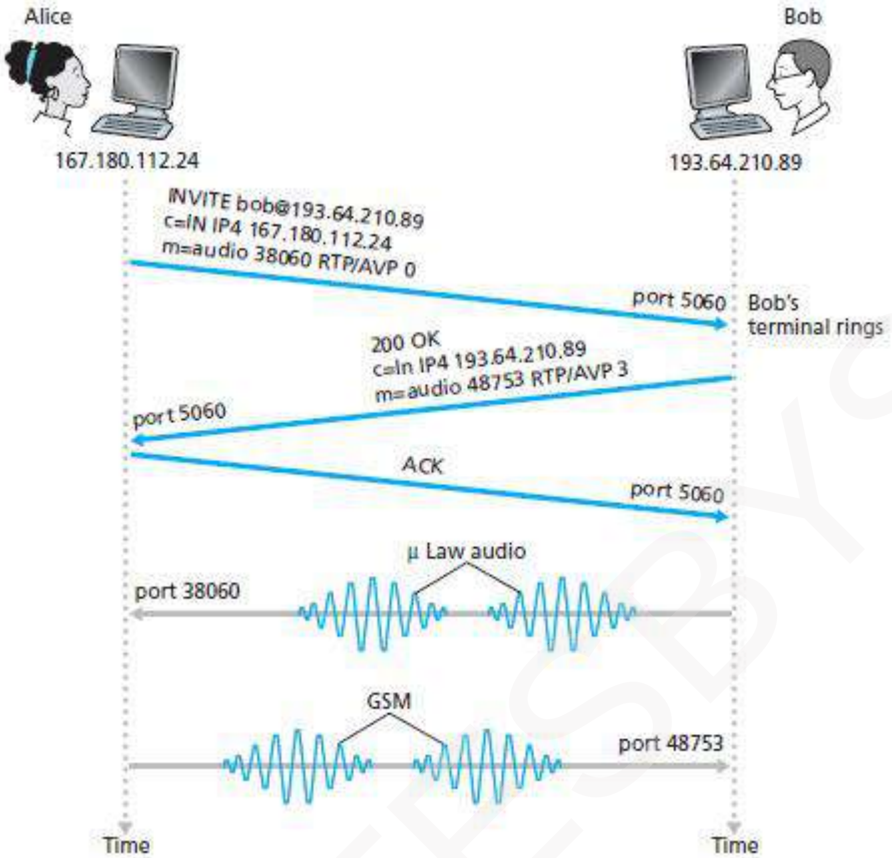

Figure 5.7: SIP call establishment when Alice knows Bob's IP address

• Consider an example: Alice wants to call Bob.
• Alice's & Bob's PCs are both equipped with SIP-based software for making and receiving phone calls.
• The following events occur:
  1) An SIP session begins when Alice sends Bob an INVITE message.
  ➢ This INVITE message is sent over UDP to the well-known port 5060 for SIP.
  ➢ The INVITE message includes
    i) An identifier for Bob (bob@193.64.210.89)
    ii) An indication of Alice's current IP address
    iii) An indication that Alice desires to receive audio, which is encoded in format AVP 0.
  2) Then, Bob sends an SIP response message (which resembles an HTTP response message).
  ➢ The response message is sent over UDP to the well-known port 5060 for SIP.
  ➢ The response message includes
    i) 200 OK
    ii) An indication of Bob's current IP address
    iii) An indication that Bob desires to receive audio, which is encoded in format AVP 3.
  3) Then, Alice sends Bob an SIP acknowledgment message.
  4) Finally, Bob and Alice can talk.
• Three key characteristics of SIP:
  1) SIP is an out-of-band protocol
  ➢ The SIP message & the media-data use different sockets for sending and receiving.
  2) The SIP messages are ASCII-readable and resemble HTTP messages.
  3) SIP requires all messages to be acknowledged, so it can run over UDP or TCP.

5.4.2.2 SIP Messages

• Suppose that Alice wants to initiate a VoIP call to Bob.
• Then, her message will look something like this:

```
L1)  INVITE sip:bob@domain.com SIP/2.0
L2)  Via: SIP/2.0/UDP 167.180.112.24
L3)  From: sip:alice@hereway.com
L4)  To: sip:bob@domain.com
L5)  Call-ID: a2e3a@pigeon.hereway.com
L6)  Content-Type: application/sdp
L7)  Content-Length: 885
L8)  c=IN IP4 167.180.112.24
L9)  m=audio 38060 RTP/AVP 0
```

• Line by line explanation is as follows:

L1) The INVITE line includes the SIP version.
L2) Via header indicates the IP address of the SIP device.
L3) Similar to an e-mail message, the SIP message includes a From header line.
L4) Similar to an e-mail message, the SIP message includes a To header line.
L5) Call-ID uniquely identifies the call.
L6) Content-Type defines the format used to describe the message-content.
L7) Content-Length provides the length in bytes of the message-content.
L8) A carriage return followed by line feed.
L9) The message contains the content.

5.4.2.3 Name Translation & User Location

• IP addresses are often dynamically assigned with DHCP.
• Suppose that Alice knows only Bob's e-mail address, bob@domain.com
• In this case, Alice needs to obtain the IP address of the device associated with the bob@domain.com.
• How to find IP address?

1) Alice creates & sends an INVITE message to an SIP proxy.
2) The proxy responds with an SIP reply.
➢ The reply includes the IP address of the device associated with the bob@domain.com.

• How can the proxy server determine the current IP address for bob@domain.com?

Answer: Every SIP user has an associated registrar.

➢ First, a user launches an SIP application on a device.
➢ Then, the application sends an SIP register message to the registrar.
➢ Finally, IP address of the device will be registered in the registrar .

• The user's registrar keeps track of his current IP address.
• When the user switches to a new device, IP address of new device will be registered in the registrar.
• The registrar is analogous to a DNS authoritative name server:

1) The DNS server translates fixed host names to fixed IP addresses;
2) The SIP registrar translates human identifiers (ex: bob@domain.com) to dynamic IP address.

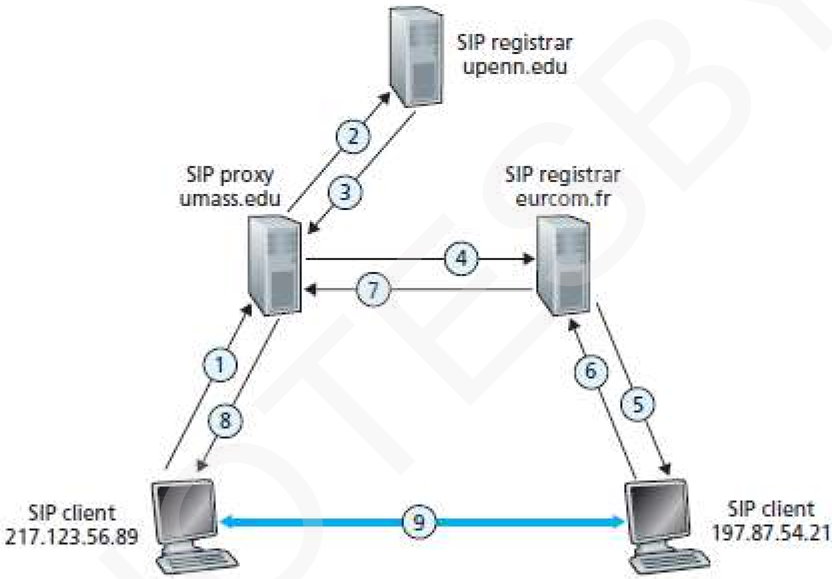

Figure 5.8: Session initiation, involving SIP proxies and registrars

• Session initiation process is illustrated in Figure 5.8.
• jim@umass.edu(217.123.56.89) wants to initiate VoIP session with keith@upenn.edu (197.87.54.21)
• The following steps are taken:

1) Jim sends an INVITE message to the umass SIP proxy.
2) The proxy
→ performs DNS lookup on the SIP registrar upenn.edu and
→ forwards then the message to the registrar server upenn.
3) keith@upenn.edu is not registered at the upenn registrar.
➢ Therefore, the upenn registrar sends a redirect response to umass proxy.
4) The umass proxy sends an INVITE message to the eurecom SIP registrar.
5) The eurecom registrar
→ knows the IP address of keith@eurecom.fr and
→ forwards INVITE message to the host 197.87.54.21 which is running Keith's SIP client.
6–8) An SIP response is sent back through registrars/proxies to SIP client on 217.123.56.89.
9) Media is sent directly between the two clients.