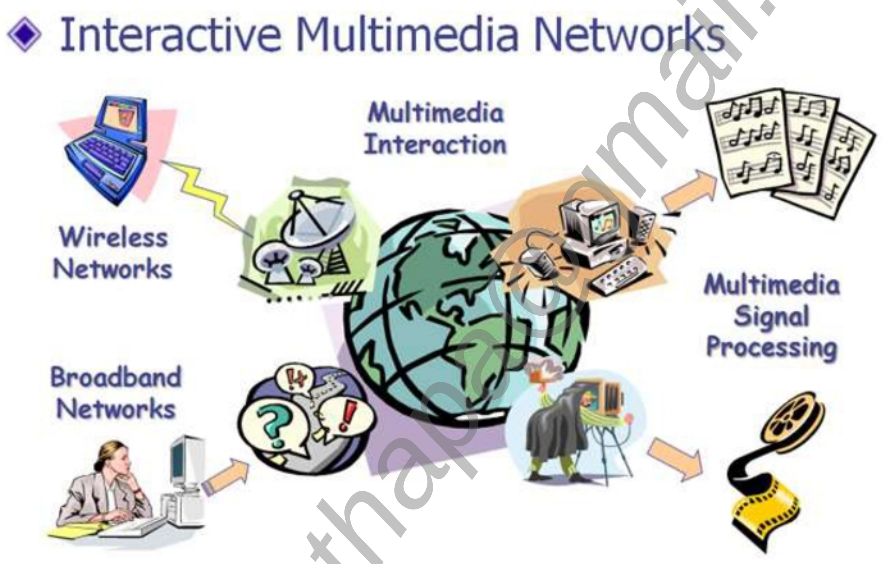
Chapter 8
Multimedia Networking Application

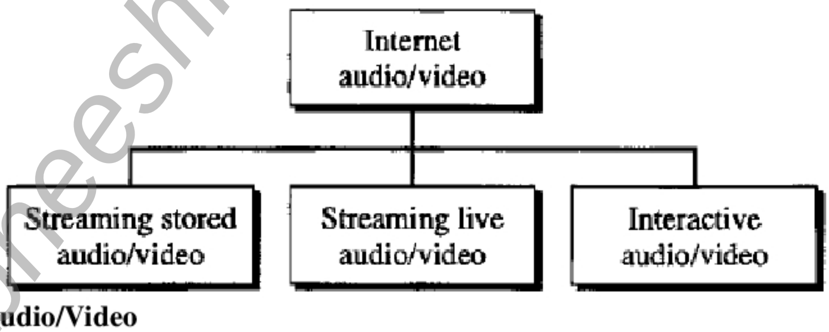
Introduction

Multimedia is a technology that enable humans to use computers capable of processing textual data, audio and video, still pictures and animations. The importance of communications or networking for multimedia lies in the new applications that will be generated by adding networking capabilities to multimedia computers. Multimedia means use of more than one media such as the combination of text, sound, audio motion video. In the past, we listened to an audio broadcast through a radio and watched a video program broadcast through a TV. We used the telephone network to interactively communicate with another party. But times have changed. People want to use network not only for text and image communications, but also for audio and video services. Networking applications whose data contains audio and video content are referred to as multimedia networking applications or continuous media application. The characteristics of multimedia make heavy demands on storage and transmission systems. Data compression can be used to reduce the demands of multimedia, particularly of video and audio on these systems, but usually at the expense of some loss in the detail compared with the source and at extra cost. Multimedia networking applications are highly sensitive to delay but are typically loss tolerant whereas static content applications are delay tolerant but loss intolerant. Applications range over entertainment, education, information provision, design e.g. CAD/CAM, co-operative working such as video conferencing, application sharing, remote working and virtual reality experiences


◆ Interactive Multimedia Networks
Multimedia Interaction
Wireless Networks
Multimedia Signal Processing
Broadband Networks

Example of Multimedia Application

We can classify multimedia application i.e. audio and video services into three broad classes.



1. **Streaming Stored Audio/Video**

   In streaming stored audio/video, the files are compressed and stored on a server. A client downloads the files through the Internet. This is sometimes referred to as on-demand audio/video. Examples of stored audio files are songs, books on tape, and famous lectures. Examples of stored video files are movies, TV shows, and music video clips.

2. **One to many streaming of live Audio/Video**

   In streaming live audio/video, a user listens to broadcast audio and video through the Internet. A good example of this type of application is the Internet radio. Some radio stations broadcast their programs only on the Internet; many broadcast them both on the Internet and on the air. Internet TV is not popular yet, but many people believe that TV stations will broadcast their programs on the Internet in the future. Streaming live audio/video refers to the broadcasting of radio and TV programs through the Internet.

3. **Interactive Audio/Video**

   In interactive audio/video, people use the Internet to interactively communicate with one another. A good example of this application is Internet telephony and Internet teleconferencing. Interactive audio/video refers to the use of the Internet for interactive audio/video applications.

Note: Streaming means a user can listen to (or watch) the file after the downloading has started.

Streaming Stored Audio/Video

Today audio/video streaming has become most popular. One of the main problem is it consume large bandwidth and large storage. In audio/video streaming, clients request compressed audio/video files, which are resident on servers. Upon client request, the server directs an audio/video file to the client by sending the file into a socket. Both TCP and UDP socket connections are used in practice. Before sending the audio/video file into the network, the file may is segmented, and the segments are typically encapsulated with special headers appropriate for audio/video traffic. Real-Time Protocol (RTP) is used for encapsulating the segments. Once the client begins to receive the requested audio/video file, the client begins to render the file within a few seconds. Most of the existing products also provide for user interactivity, e.g., pause/resume and temporal jumps to the future and past of the audio/video file. User interactivity also requires a protocol for client/server interaction. Real Time Streaming Protocol (RTSP), is used for providing user interactivity.

Accessing Audio/Video from Server

The stored audio/video can either reside on a Web server, or on an audio/video streaming server.

Consider first the case of audio/video streaming.

When an audio file resides on a Web server, the audio file is an ordinary object in the server's file system, just as are HTML and JPEG files. When a user wants to hear the audio file, its host establishes a TCP connection with the Web server and sends an HTTP request for the object ; upon receiving such a request, the Web server bundles the audio file in an HTTP response message and sends the response message back into the TCP connection. The case of video can be a little more tricky, only because the audio and video parts of the "video" may be stored in two different files, that is, they may be two different objects in the Web server's file system. In this case, two separate HTTP requests are sent to the server (over two separate TCP connections for HTTP/1.0), and the audio and video files arrive at the client in parallel. It is up to the client to manage the synchronization of the two streams. Let us assume that the audio and video is contained in one file.

A naive architecture for audio/video streaming is shown in Figure below this architecture:

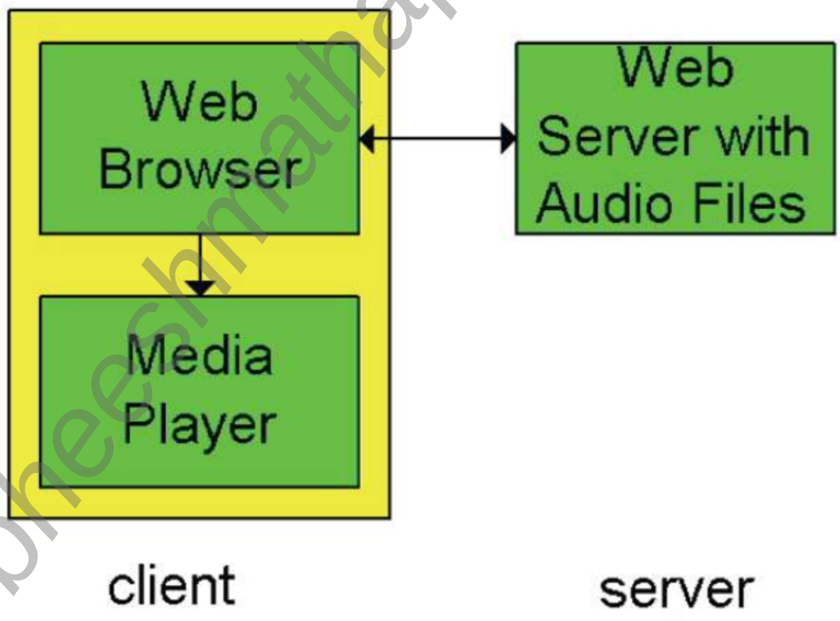


Figure 6.2-1 A naive implementation for audio streaming.

1. The browser process establishes a TCP connection with the Web server and requests the audio/video file with an HTTP request message.
2. The Web server sends to the browser the audio/video file in an HTTP response message.
3. The content-type: header line in the HTTP response message indicates a specific audio/video encoding. The client browser examines the content-type of the response message, launches the associated media player(helper application), and passes the file to the media player.
4. The media player then renders the audio/video file.

Second Case of Streaming Audio/Video

The above architecture is very simple however it has a major drawback i.e. the media player must interact with the server through the intermediary of a Web browser. This can lead to many problems. In particular, when the browser is an intermediary, the entire object must be downloaded before the browser passes the object to a helper application; the resulting initial delay is typically unacceptable for audio/video clips of moderate length. For this reason, audio/video streaming implementations typically have the server send the audio/video file directly to the media player process. In other words, a direct socket connection is made between the server process and the media player process. As shown in Figure below.

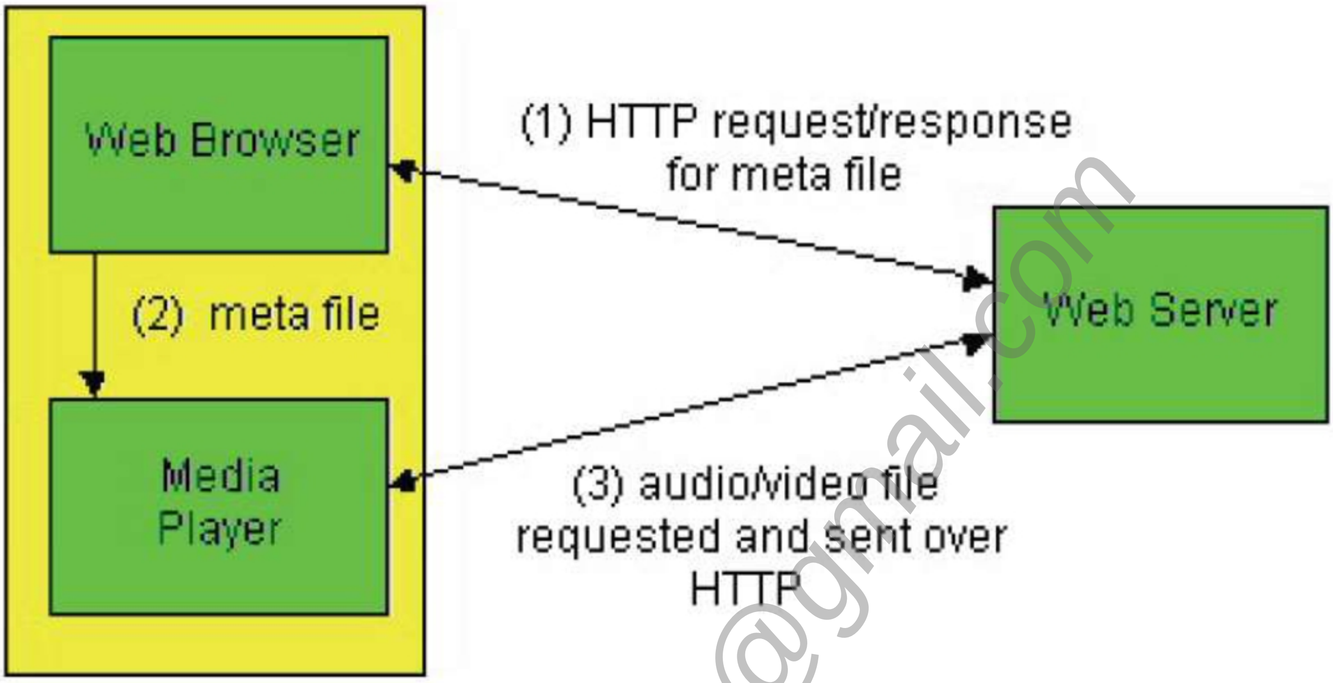

Web Browser
(1) HTTP request/response for meta file
(2) meta file
Web Server
Media Player
(3) audio/video file requested and sent over HTTP

Figure 6.2-2 Web server sends audio/video directly to the media player.

This is done by making use of a **meta file**, which is a file that provides information (e.g., URL, type of encoding, etc.) about the audio/ video file that is to be streamed.

A direct TCP connection between the server and the media player is obtained as follows:

1. The user clicks on a hyperlink for an audio/video file.
2. The hyperlink does not point directly to the audio/video file, but instead to a meta file. The meta file contains the the URL of the actual audio video file. The HTTP response message that encapsulates the meta file includes a `content-type:` header line that indicates the specific audio/video application.
3. The client browser examines the content-type header line of the response message, launches the associated media player, and passes the entity body of the response message (i.e., the meta file) to the media player.
4. The media player sets up a TCP connection directly with the HTTP server. The media player sends an HTTP request message for the audio/ video file into the TCP connection.
5. The audio/video file is sent within an HTTP response message to the media player. The media player streams out the audio/video file.