Multimedia Streams
Synchronization
• Synchronization in multimedia systems refers to the temporal relations between media stream data units in multimedia system.

• Time dependent media data units: Temporal relations between consecutive units of media stream exist. If the presentation durations of all units of a time-dependent media object are equal, it is called continuous media object. e.g Ordered frames of video has fixed presentation time.

• Time independent media data unit is any kind of traditional media like text and image. Semantic of respective content does not depend upon a presentation according to the time domain.
Alignment of entities regarding

Content:

Data dependencies: e.g. different views of the same data
Dependency between spreadsheet and graphics that represent data listed in spreadsheet.

Space

Layout relationships: Desktop publishing expressed using layout frames and a content is assigned to this frame. Positioning of a layout frame in a document may be fixed to a position in a document to a position on a page. In window based system, layout frames correspond to windows and video can be positioned in window.

Time

Layout in time: e.g. simultaneous playout of audio & video
Eg Lip Synchronization.
Temporal synchronization

The process of maintaining the temporal order of one or several media streams

Relations relevant for audio and video exchange:

*Intra-stream synchronization:* Voice samples and video frames need to arrive in time at the receiver before display or playout time to maintain the continuity of playback.

*Inter-stream synchronization* is needed to present voice and video samples in a certain relation.

*Spatial synchronization:* Participants in a multimedia conference should receive audio-visual data at the same time although geographically distributed.
Synchronization considered at different levels since it is supported by several components...

Synchronization considered at several levels of Multimedia Systems

**Level 1:** OS and lower level communication layers, CPU scheduling, semaphores during IPC, traffic shaping network scheduling

**Objective:** avoid jitter at presentation time of one stream

**Level 2:** Middleware/Session layer (Run-time), Synchronization of multimedia streams (schedulers)

**Objective:** bounded skews between various streams

**Level 3:** Application layer (Run-time), Support for synchronization between time-dependent and time-independent media together with handling of user interaction

**Objective:** bounded skews between time-dependent and time-independent media
Synchronization Tools
Classification of a system as a MM system

System can be distinguished by:

Number of media

Types of supported media

Degree of media integration

Combining all three criteria, a MM system or application that supports the integrated processing of several media types with at least one time dependent medium.
Classification of Media Use

The arrows indicate the increasing degree of MM capability for each Criterion.

Integrated digital system support all types of media. Hybrid systems that handle time dependent analog media and time independent digital media.
Synchronization Specification

Implicit

Temporal relation specified implicitly during capturing of media objects

Goal: use this temporal relation to present media in the same way as they were originally captured

Example: Audio and Video recording and playback
Explicit

Temporal relation specified explicitly to define dependency in case media objects were created independently

Example: creation of slide show, Presentation designer selects slides, creates audio objects, defines units of audio presentation stream, defines units of audio presentation stream where slides have to be presented
Sources of Asynchrony

- **Different delays**: Assumption of independent network connections imposes different delays.
- **Network jitter**: asynchronous data transfer destroys synchrony.
- **End-system jitter**: packetizing and depacketizing of media data units with different size due to encoding introduces jitter as well as passing media units through the lower protocol layers.
- **Clock drift**: between the clocks in the servers and in the client is present because we do not assume global clocks.
- **Change of the average delay**: the synchronization scheme has to be adaptive with respect to a change of the average delay.
- **Server drop outs**: due to process scheduling are a realistic assumption when using non-real-time operating systems.
Types of Synchronization

- **Time based synchronization**

- To synchronize and to disseminate multimedia data using time as a major parameter

- Synchronization is achieved by mean of electronic time signals generated ad hoc by devices directly connected to a primary time standard and transmitted by radio or by cable.

- Time signals classes: video or sound signals (called right time signals) for low accuracy manual synchronization, digital signals for automatic synchronization with medium-high accuracy

- Passive synchronization techniques: Do not use a time signal generated ad hoc, instead it is used a signal generated for other purpose
Types of Synchronization

• *Delay based synchronization*
  - Schemes compensate for these delays by computing well-defined starting times for each stream server of the application.
  - Allow to initiate the synchronized playback of a media stream that is composed of several multimedia streams.

• *Jitter based synchronization*
  - Jitter is experienced by media units traveling from the source to the destination.
  - To smoothen out jitter, elastic buffers are required.
  - These schemes guarantee a smooth playback of the stream
  - Very low buffer requirements.
Synchronization Classification

- **Intra object Synchronization:**
- **Time relation between various presentation units of one time-dependent media stream**
- e.g. time relation between single frames of a video sequence. For video with rate of 25 frames/sec each of the frame must be displayed for 40ms.

Audio and video streams are isochronous in nature. processing and network delay jitter (i.e., the variance in delay); variations in rates of recording and playback; and unreliable transmission of stream data units.

Solutions:

- buffer monitoring
- feedback technique
- global clock
- Inter-object Synchronization
- Time relation between media objects

- Multiplexing of streams
- Aggregation in one data structure
- Global clocks
- Synchronization marker
- Synchronization channel
- Feedback technique
Spatial Synchronization

- All participants in the conference receive the audio and video data at the same time, to maintain a fair conference.
- Based on global clocks synchronization channel or feedback techniques as proposed for inter-stream synchronization.
- When global clocks are available, mechanisms based on these clocks can achieve the most accurate spatial synchronization.
Gap Problem in Synchronization

What does blocking of stream mean for output device?

- Should we repeat previous music, speech, picture?
- How long should such gap exist?
- **Solution 1: restricted blocking method**
  - Switch output device to last picture as still picture
  - Switch output device to alternative presentation if gap between late video and audio exceeds predefined threshold
- **Solution 2: resample stream**
  - Speed up or slow down streams
- **Off line re-sampling** used after capturing of media streams with independent streams, Example: concert which is captured with two independent audio/video devices. Online re-sampling used during presentation in case gap between media streams occurs
Synchronization Handlers

• **roles**: to identify the stakeholders involved, as well as the **prohibition** and **policy** concepts.

• **Roles of these components in handling synchronization**
  - Service provider
  - Network provider
  - End-user roles
Service Provider

- **Intra-stream synchronization**: SP manipulates incoming audio/video streams so that outgoing streams are within the 10 ms jitter boundary;

- **Inter-stream synchronization**: SP will manipulate incoming audio/video streams so that related outgoing audio and video streams are within the -20 ms and +40 ms range;

- **Spatial synchronization**: SP is responsible for ensuring that outgoing audio/video streams are played out simultaneously at the multiple users within the 0.25 s boundary.
Network Provider

- **Compulsory network service:** a transportation service with a deterministic service is provided

- **Statistical reliable network service:** a transportation service with a certain percentage of QoS violations is provided.

- **Best effort network service:** the request from a client for a certain transport service is evaluated against the current network traffic.
End-User

- With respect to the display of audio and video it is important that the following requirements to be met:
  - lip-synchronization is a well known requirement and should be in the -20 ms to +40 ms range;
  - audio or video jitter should be within the range of 10 ms;
  - loss of video frames or audio samples is tolerable when less than 1% of the total sent;
  - spatial synchronization should be in the range of -0.25 s to +0.25 seconds.
End-User

• **compulsory end-user service**: in this case the synchronization requirements must be met.

• **statistical reliable end-user service**: a certain percentage of violations of the synchronization requirements is allowed.

• **best effort end-user service**: possibilities to fulfill the synchronization requirements are based on current processing and storage activities.

• User policies are often application dependent.
Synchronization Classification

- Live Synchronization

  - **Without storage**
    - Capture data with timing at source
    - Transmit data with timing information
    - Reconstruct timing during playout (with delay)
    - May modify only data parameters (e.g. format, ...)

    ![Diagram of without storage synchronization]

  - **With storage**
    - Similar, decoupling allows for also
      - Changing speed, delay, ...
      - Random access

    ![Diagram of with storage synchronization]
Synthetic Synchronization

- **Goal:** arrange stored data objects to provide new combined multimedia objects via artificial temporal relations

- **Requirements:** support flexible synchronization relations between media

- **Example:** authoring, tutoring systems

- **Two phases:**
  - **Specification phase:** define temporal relations
  - **Presentation phase:** present data in sync mode

- **Example:** 4 audio messages recorded related to parts of engine in animation. Animation sequence shows a slow 360 degree rotation of engine
Logical Data Units and their Classification

- Time dependent presentation units are called logical data units (LDU)s.

- LDU classification

  - Open: Duration of Open LDU is not predictable before execution of the presentation. eg. Camera or microphone.

  - Closed LDU have predictable duration, eg. Parts of stored media objects of continuous media like A/V or stored media with fixed duration.

- LDUs important

- In specification of synchronization
• Lip Synchronization Requirements
• In sync: -80ms ≤ skew ≤ 80ms
• Out of sync: Skew < -160ms, Skew > 160ms
• Transient:
• -160ms ≤ skew < -80ms
• 80ms < skew ≤ 160ms
Synchronization Reference Model

- Four layer SRM. Each layer implements synchronization mechanism which are provided by appropriate interface that are used to enforce temporal relationships.
1. **Media Layer:** At the media layer, an application operates on a single continuous media stream, which is treated as a sequence of Media Data Units (MDUs). The abstraction offered at this layer is a device-independent interface with operations like \texttt{read(device handle, MDU)} and \texttt{write(device-handle, MDU)}. To set up a continuous media stream using the abstractions offered by the media layer, an application executes a process for each stream.

2. **Stream Layer:** The stream layer operates on continuous media streams, as well as on groups of media streams. In a group, all streams are presented in parallel by using mechanisms for inter-stream synchronization. The abstraction offered by the stream layer is the notion of streams with timing parameters concerning the QoS for intra-stream synchronization in a stream and inter-stream synchronization between streams of a group. Continuous media is seen on top of the stream layer as a data flow with implicit time constraints; individual MDU’s are not visible. The streams are executed in a real-time environment, where all processing is constrained by well-defined time specifications. On the other hand, the applications themselves that are using the stream layer services are executed in a non real-time environment, where the processing of events is controlled by the operating system scheduling policies.
3. **Document Layer:** The document layer operates on all types of media and hides the differences between time-independent and time-dependent media. The abstraction offered to the application is that of a complete, synchronized presentation. This layer takes a synchronization specification as input and is responsible for the correct schedule of the overall presentation.

The task of this layer is to close the gap between the needs for the execution of a synchronized presentation and the stream-oriented services. The functions located at the document layer are to compute and execute complete presentation schedules that include the presentation of the noncontinuous media documents and the calls to the stream layer. Furthermore, the document layer is responsible for initiating preparative actions that are necessary for achieving a correctly synchronized presentation. The document layer does not handle the inter-stream and intra-stream synchronization. For these purposes, it uses the services of the stream layer.

4. **Specification Layer:** The specification layer is an open layer. It does not offer an explicit interface. This layer contains applications and tools that allow one to create synchronization specifications. Examples of such tools are synchronization editors, multimedia document or object editors and authoring systems. Also located at the specification layer are tools for converting specifications to a document layer format. An example of such a conversion tool is a multimedia document formatter that produces an MHEG specification.
Synchronization Models

- The Temporal Synchronization Model

Diagram:
- Record U1
- Interaction
- Audio 1
- Video
- P1, P2, P3, P4
- ?
- t
At the time of creation of multimedia information, a user needs to model temporal interrelationships among various media data units and the possible options for the quality requirements which must be observed at the time of playback for each data unit. Coordinating the real-time presentation of information and maintaining the temporal relationships among component media is known as temporal synchronization.

The concept of temporal synchronization is illustrated in figure 6.3. where a sequence of video, audio, and animation clips are presented in time to compose a multimedia document. One may notice from this figure that the system must observe some timing relationships among various data units in order to present the information to the user in a meaningful way. These relationships can be live or synthetically created. A voice-annotated slide show, on the other hand, is example of a synthetically created relationship between audio and animation information. In this case, change of an image and the end of its verbal annotation represent a synchronization point in time.
Synchronization Models

- Unix Synchronization Model

![Diagram of Synchronization Models](image)
The Unix operating system has some procedures to handle the media synchronization problems. The facilities provided by UNIX for allowing the close synchronization of even two streams are limited by the processing granularity provided to an application layer.

Unix uses following to support multimedia synchronization processing:

- **Synchronization classes**: UNIX can do reasonably well in supporting serial synchronization of data if the sampling rates are sufficiently low to not cause a burden on the system. The block-oriented fetching of data can significantly increase the number of samples processed by an application, although the limited scheduling control of each thread will not ensure the constancy required by high-bandwidth devices. For parallel synchronization, the prospects are less promising: the sequential nature of UNIX I/O will result in either a loss of data resolution or in a limit on the number of parallel tracks that can be processed. One reason for this is the form of the generic I/O system call; all I/O is done on a single file descriptor at a time, with separate file descriptors requiring separate system calls. It may be possible to build multiplexing drivers to combine I/O on a number of file descriptors, but this will not offer a general solution to most applications builders. Another possibility may be the development of multi-file I/O system calls, but even if these were to become accepted by the growing list of standards organizations, most languages would be unable to cope with the notions of parallel I/O accesses. For the time being, the best one can hope to do is to provide either an applications-based multi-threaded scheduling solution to parallel stream synchronization or to rely on smarter controllers to by-pass the CPU altogether.
- **Synchronization scope:** Of the two types of synchronization scope, point synchronization can be relatively well-managed by the thread level, but continuous synchronization can only be managed if the input and output data rates are sufficiently low. Once again, the scope of the synchronization is not only restricted by the implementation concerns of the UNIX I/O system, but also by the ability of applications code to flexibly access data at a low-enough layer in the system.

- **Synchronization masters:** the easiest way to support synchronization within a UNIX environment is to have a master clock regulate the gathering of samples and the dispatching of samples to various output devices. In order for such a clock to function, it will need to be able to influence processing in a number of threads in the same way that real-time clock can influence the scheduling of various real-time processes. Unfortunately, the level of real-time support in UNIX systems has never been particularly good. As for peer-level synchronization, the problems with guaranteed scheduling time under UNIX once again limit the amount of coordinated processing that can be realistically accomplished.

- **Synchronization precision:** depending on the level of precision, processing can be implemented at any of the five layers in the UNIX hierarchy. If stereo channels need to be synchronized, then it can only occur at the controller or interrupt level (unless the data need only be resynchronized at a much lower rate). If, on the other hand, subtitles need to be added to a running video sequence, then this can easily be done at the thread level.