Notes for week 5 part 1 (MPEG Continuity)

MPEG Streaming

An ISO 11172 (MPEG-1) stream runs at 1.2Mbps. This contains both the audio and video portions of the stream. For the audio stream, MPEG-1 uses 48k samples/second across two channels (stereo left, stereo right) and 2-bits per sample. From this, we can determine the bandwidth that audio in an MPEG stream consumes:

Audio B/W = 2 channels x 48k samples/second x 2 bits/sample = 192kbps

Since we know that the total bandwidth is 1.2Mbps, we can now determine the video bandwidth to be:

Video B/W = 1.2Mbps – 192kbps = 1008kbps

Staggered Toggling with MPEG Packs

The maximum duration of an MPEG pack is defined to be 0.7 seconds. From the audio and video rates above, we can now determine the bits per pack for each stream:

Audio bpp ≤ 0.7s x 192kbps = 134.4kbpp
Video bpp ≤ 0.7s x 1008kbps = 705.6kbpp

Given a packet length of 188 bytes (MPEG-2), the maximum number of audio and video packets per pack can be determined:

Audio packets per pack ≤ 134.4kbpp/(188 bytes * 8-bits/byte) = 89 packets/pack
Video packets per pack ≤ 705.6kbpp/(188 bytes * 8-bits/byte) = 469 packets/pack

Now that the number of whole audio and video packets has been determined, we need to check the duration of each to see that they are the same (otherwise, the audio/video will slowly drift out of synchronization).

Audio duration = \( \frac{89 \text{ packets/pack} \times 188 \text{ bytes} \times 8 \text{ bits/byte}}{192000 \text{ bps}} = 697 \text{ ms} \)

Video duration = \( \frac{469 \text{ packets/pack} \times 188 \text{ bytes} \times 8 \text{ bits/byte}}{1008000 \text{ bps}} = 699 \text{ ms} \)

Notice that there’s a 2ms difference between the two. To account for this, a technique called staggered toggling is used. First, we need to pick one of the two streams as our “master” (i.e. the stream who’s packets/pack will stay the same). For this case, we’ll pick the audio stream.
Using the audio stream as the master, the ratio of the video/audio data rates is needed. This is computed by:

\[
\frac{1008 \text{ kbps}}{192 \text{ kbps}} = 5.25
\]

This ratio indicates that for every 1 packet of audio, exactly 5.25 packets of video are needed. So, with 89 audio packets/pack:

\[
5.25 \times 89 \text{ packets/pack} = 467.25 \text{ video packets are needed.}
\]

Since it’s not possible to encode a quarter of a packet, every fourth pack in the stream will contain one extra video packet to compensate. This is called staggered toggling.

Ex:

PACK 1: 89 Audio Packets, 467 Video Packets  
PACK 2: 89 Audio Packets, 467 Video Packets  
PACK 3: 89 Audio Packets, 467 Video Packets  
PACK 4: 89 Audio Packets, 468 Video Packets  
PACK 5: 89 Audio Packets, 467 Video Packets  
PACK 6: 89 Audio Packets, 467 Video Packets  
PACK 7: 89 Audio Packets, 467 Video Packets  
PACK 8: 89 Audio Packets, 468 Video Packets  
...
Continuity of MPEG Playback over a network

When transmitting an MPEG stream over a packet switched network, there are 5 potential problems for continuity of the stream:

1) network delay
2) transmission errors
3) non-sequential delivery of MPEG data
4) bandwidth issues
5) jitter (variation in network delay)

The major problem for MPEG streaming is the jitter variation, which will be handled by buffering. The buffer must be large enough to handle worst case situations, which will be described using the following model:

Where:

- \( a = \text{SCR}_p - \text{SCR}_1 \) (amount of time from when the 1st pack was sent, to p\text{th} pack)
- \( b \) = network delay experienced by pack p
- \( c \) = network delay experienced by pack 1
- \( x \) = amount of time from when the 1st pack was received, to the time the p\text{th} pack was received

The network delay is bounded by a maximum delay (\( \Delta_{\text{max}} \)), and a minimum delay (\( \Delta_{\text{min}} \)). By switching the delays for \( b \) and \( c \), two cases emerge that need to be accounted for to guarantee continuity of playback: starvation, and overflow. The following table describes each situation:

<table>
<thead>
<tr>
<th>Situation</th>
<th>Starvation</th>
<th>Overflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( \text{SCR}_1 - \text{SCR}_p )</td>
<td>( \text{SCR}_1 - \text{SCR}_p )</td>
</tr>
<tr>
<td>( b )</td>
<td>( \Delta_{\text{max}} )</td>
<td>( \Delta_{\text{min}} )</td>
</tr>
<tr>
<td>( c )</td>
<td>( \Delta_{\text{min}} )</td>
<td>( \Delta_{\text{max}} )</td>
</tr>
<tr>
<td>( x )</td>
<td>( a + (b - c) )</td>
<td>( a + (c - b) )</td>
</tr>
</tbody>
</table>

In order for the buffer to be able to handle both situations, it must first buffer (\( \Delta_{\text{max}} - \Delta_{\text{min}} \)) worth of data to avoid starvation. It will also need to be able to buffer an additional (\( \Delta_{\text{max}} \)).
- \( \Delta_{\text{min}} \) worth of data to avoid overflow. Which means that in total, the buffer size must be of sufficient size to hold \([2 \times (\Delta_{\text{max}} - \Delta_{\text{min}})]\) worth of MPEG data.