Technical Disclosure Commons

Defensive Publications Series

May 09, 2017

In-Band Speed Test for Livestreams

Brian Schmidt

IN-BAND SPEED TEST FOR LIVESTREAMS

Livestreaming includes capturing of live media and transmitting the live media by a user device over a network to a remote server or another user device. For example, a mobile device can capture live video as a video stream and live audio as an audio stream and transmit the live video stream and the live audio stream over a network to a remote ingestion server. The network may be a wireless network, a cellular data network, a mobile hot spot, etc. The network has network conditions, such as available bandwidth and latency to the remote server, that affect the livestreaming.

For the media to be transmitted live (e.g., without delay), various parameters of the livestream (e.g., video resolution, encoder bitrate, frame rate, etc.) can be set based on the network conditions. Some of the parameters are to be set prior to initiating livestreaming (e.g., setting the resolution to high definition or standard definition, etc.) instead of during livestreaming. If the network conditions are too poor to support a reasonable livestream broadcast, the user device (e.g., via a host application) can disable livestreaming to prevent users from a bad livestreaming experience.

Typically, livestreaming begins by a user device negotiating one or more parameters of a network connection with a server device. The negotiating of the one or more parameters does not include sending of actual live media and does not account for the size of the live media. The negotiating of the one or more parameters may include a typical network speed test (e.g., estimating download and upload speed by transmitting a sample file to the user device with instructions for the user device to send the file back). After negotiating the one or more parameters, the user device opens a network connection for transmitting a livestream, captures live media per the one or more parameters (e.g., determined per the typical network speed test), encapsulates the live media in virtual channels (e.g., per livestreaming protocol), multiplexes the virtual channels to generate multiplexed data, and transmits the multiplexed data over the network connection to the server device. Per livestreaming protocol, video data is encapsulated within one or more video virtual channels with a video data identifier, audio data is encapsulated within one or more audio virtual channels with an audio data identifier, and data (e.g., metadata, etc.) is encapsulated within one or more data virtual channels. The data identifiers (e.g., channel IDs) may be known *a priori* based on the livestreaming protocol specification or may be negotiated as part of the stream setup (e.g., negotiating parameters of the network connection). The server device receives the multiplexed data, demultiplexes the multiplexed data and directs, based on the data identifiers, audio data to an audio decoder and video data to a video decoder. The server device discards data in unused virtual channels (e.g., one or more of the data virtual channels).

The negotiating of the one or more parameters of the network connection prior to livestreaming does not include sending of actual live media, does not account for the size (e.g., bitrate) of the actual livestream to be transmitted, and does not account for the capacity of the user device. The one or more negotiated parameters may not be an accurate representation of available bandwidth for the sending of the actual live media. Live media captured based on the one or more negotiated parameters may be captured at a quality lower than the available bandwidth allows or may be captured at quality that is so high that it causes delay.

Negotiating of the one or more parameters during livestreaming has side effects including one or more of a poor viewing experience (e.g., changing of quality of the livestream may be perceivable by users, a user may stop viewing the livestream because of the changing of quality, etc.), the incoming live media will need to be processed (e.g., via the ingestion server, via the receiver, etc.), and the types of adjustments to quality may be limited (e.g., changing of the resolution of the livestream may not be permitted during livestreaming, etc.).

The above and other deficiencies are addressed by providing an in-band speed test for livestreams to determine the network conditions (e.g., actual network conditions for the actual livestream transmitted by the actual user device). The in-band speed test for livestreams includes sending of actual live media, accounting for size of the livestream to be transmitted, and accounting for capacity of the user device. The in-band speed test includes encapsulating audio data and video data in virtual channels with data identifiers that are not for audio or video (e.g., unused virtual channels) so that actual network conditions of the real stream are tested, but the server device or any unmodified receiver will simply ignore (e.g., discard) the incoming audio and video data, thereby there are no side effects of running the test. The in-band speed test does not have the side effects of changing quality of the livestream during livestreaming, does not require processing of the live media (e.g., the live media is discarded by the server device or receiver), and can include adjustments that can only be made before livestreaming (e.g., changing of resolution of the livestream). Once the test is complete, the audio and video streams are assigned to proper virtual channels and the stream proceeds as normal. The in-band speed test for livestreams uses the true network connection to test network conditions, resulting in an accurate result. The in-band speed test also works with any standard receiver that supports a streaming protocol.

FIGS. 1 and 2 depict flow diagrams for illustrative examples of methods for implementing an in-band speed test for livestreams. The method in FIG. 1 is an example method from the perspective of a user device that is to transmit the livestream. The method in FIG. 2 is an example method from the perspective of a device (e.g., server device, second user device) that is to receive the livestream. The methods may be performed by processing devices that may include hardware, software, or a combination of both. FIGS. 1 and 2 refer to video data and audio data, but may be applied to any type of live media.

Referring to FIG. 1, a processing device of a user device opens a network connection for transmitting a video item as a livestream. Prior to opening of the network connection, the user device may negotiate the connection parameters with the device (e.g., server device, second user device) that is to receive the transmission of a video item as a livestream.

The processing device captures video data (e.g., one or more video frames, a video stream) and audio data (e.g., audio, an audio stream). The processing device may capture the video data and audio data based on the negotiated connection parameters (e.g., bandwidth, upload speed, etc.). The processing device may capture the video data and the audio data at a high quality (e.g., one of the highest capturing capacities of the user device). The video data and audio data may not be part of the livestream, but may be captured at a livestream quality that is to be tested.

The processing device encapsulates the video data and the audio data in one or more unused virtual channels (e.g., the video data in a first unused virtual channel and the audio data in a second unused virtual channel, the video data and audio data in the same unused virtual channel). Each unused virtual channel may have a data identifier that is not for audio data or video data. Typically, metadata is encapsulated in an unused virtual channel, video data is encapsulated in video virtual channels, and audio is encapsulated in audio virtual channels. The different video virtual channels and audio virtual channels may be at different qualities, have different formatting, etc. One or more of the video virtual channels and one or more of the audio virtual channels may be discarded or may not be received by the server device. One or more of the video virtual channels and audio virtual channels (e.g., that are received and not discarded by the server device) may be processed (e.g., for playing via a second user device). In FIG. 1, the video data and audio data are encapsulated in one or more unused virtual channels so that the server device receives the video data and audio data and discards the video data and audio data without having the video data and audio data processed (e.g., transmitted to a second user device, played, etc.).

The processing device transmits, over the network connection, the one or more unused virtual channels as stream data. Typically, buffering is used to smooth out disruptions in transmitting stream data. Buffering may be disabled so that the processing device can determine disruptions during transmitting of the stream data to determine network conditions.

The processing device determines network conditions based on transmission of the stream data. The capturing, encapsulating, and transmitting may be repeated over a period of time to determine the network conditions. The network conditions may include available bandwidth of the network and capacity of the processing device (e.g., local processing overhead, etc.). The quality of the stream data may be adjusted during the determining of the network conditions to determine a quality that matches the network conditions (e.g. the user device may begin transmitting stream data at 10 megabits per second (Mb/s) and may adjust down to an available 1 Mb/s).

The processing device sets quality of the video item to be transmitted as the livestream based on the network conditions. The quality of the video item may include video resolution, encoder bitrate, frame rate, etc. In FIG. 1, the quality of the video item is set by transmitting actual video and audio data (e.g., at the same bit rate at which the livestream will be transmitted) by the actual user device (e.g., the same user device that will be transmitting the video item as the livestream) over the actual network connection (e.g., over the network connection that the user device will be transmitting the video item as the livestream). The method in FIG. 1 can be utilized by a user device that supports streaming protocol. FIG. 1 may occur prior to starting a livestream broadcast.

After setting the quality of the video stream, the processing device may capture (e.g., per the quality determined in FIG. 1) a video stream and an audio stream of the video item to be transmitted as the livestream, encapsulate the video stream within a video virtual channel with a video data identifier and the audio stream within an audio virtual channel with an audio data identifier (e.g., assign the video and audio streams to the proper virtual channels per streaming protocol), multiplex the virtual channels to generate multiplexed data, and transmit the video item as the livestream including the multiplexed data over the network connection (e.g., to the server device, to a second user device, etc.).

Referring to FIG. 2, a processing device of a device (e.g., server device or second user device) may receive, over the network connection from the user device, stream data including video data and audio data in one or more unused virtual channels. In one implementation, the processing device receives a media item (e.g., one or more of video, audio, screenshots, etc.) in the one or more unused virtual channels. The media item may have been captured by the user device based on one or more parameters of a network connection (e.g., bandwidth, upload speed, etc.) that were negotiated with the processing device. The quality of a video item to be received by the processing device as a livestream may be based on network conditions determined (e.g., by the user device) based on the transmission by the user device of the stream data.

The processing device discards the video data and the audio data in the one or more unused virtual channels. The processing device may repeatedly receive and discard video data and audio data in the one or more unused virtual channels as the user device determines the network conditions. The processing device may not process (e.g., decode for playing) the video data or audio data. By the user device transmitting the video data and audio data in one or more unused virtual channels, actual network conditions may be determined and the processing device ignores the incoming video and audio data in the virtual channel (e.g., incoming video and audio data will not be processed or played). FIG. 2 may occur prior to the user device starting a livestream broadcast.

After the quality of the video item of the livestream has been set, the processing device receives the video item (e.g., the quality of the video item is based on the network conditions) as the livestream including multiplexed data over the network connection from the user device. The processing device demultiplexes the multiplexed data to obtain video data in a first virtual channel with a video data identifier and audio data in a second virtual channel with an audio data identifier, directs the video data in the first virtual channel to a video decoder based on the video data identifier, and directs the audio data in the second virtual channel to an audio decoder.

ABSTRACT

A method for in-band speed test for live streams is described. A user device opens a network connection for transmitting of a video item as a livestream, captures video data and audio data, encapsulates the video data and the audio data in one or more unused virtual channels, transmits the one or more unused virtual channels as stream data, determines network conditions based on transmission of the stream data, and sets quality of the video item to be transmitted as the livestream based on the network conditions. The server device receives the stream data including the video data and the audio data in the one or more unused virtual channels and discards the video data and the audio data in the one or more unused virtual channels.

Keywords: live stream, video, audio, ingestion server, network condition, determine, test, broadcast, channel, in-band, identifier, mobile phone, protocol, encapsulate, separate, two, multiplex, demultiplex

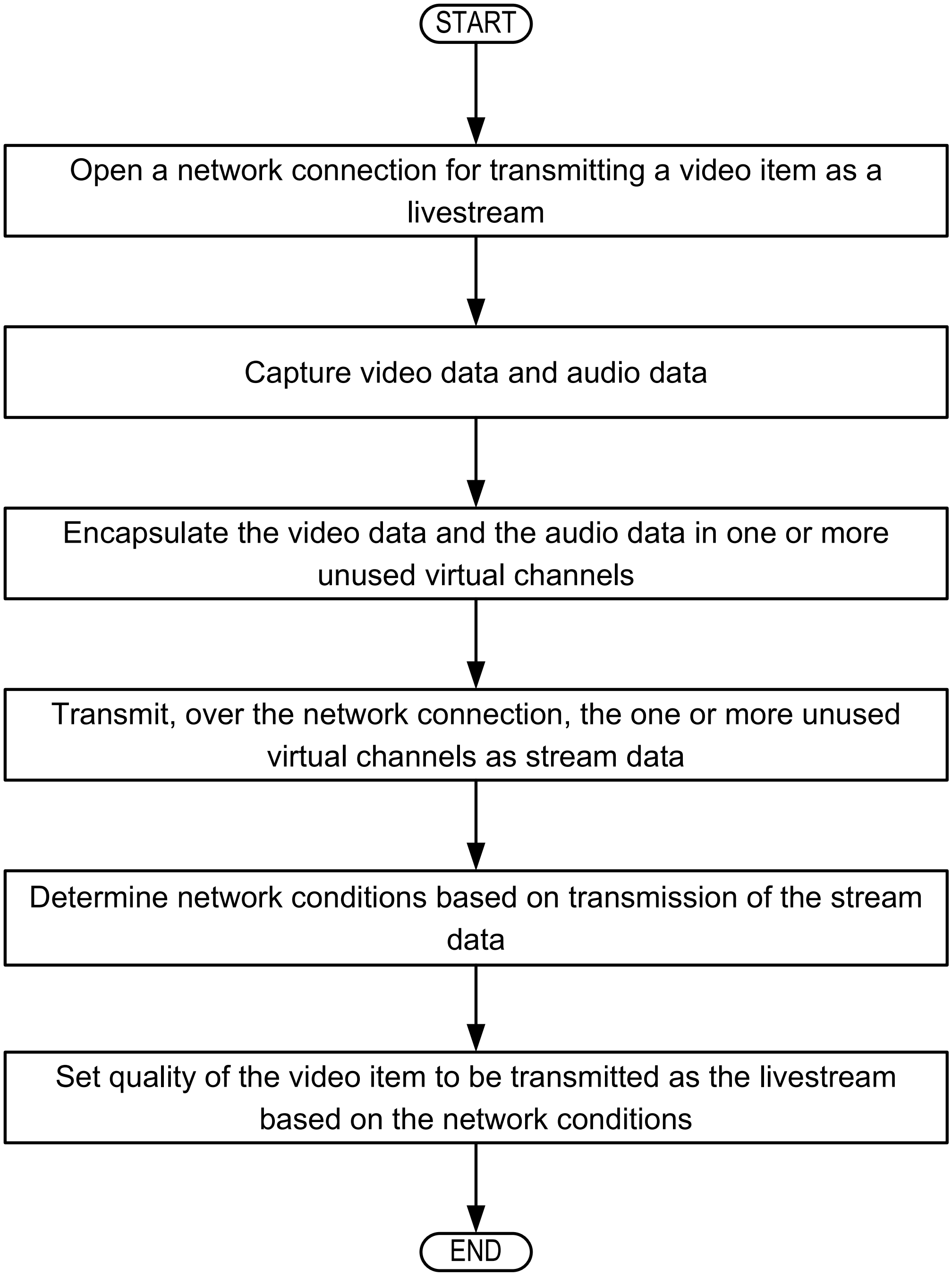
START
Open a network connection for transmitting a video item as a livestream
Capture video data and audio data
Encapsulate the video data and the audio data in one or more unused virtual channels
Transmit, over the network connection, the one or more unused virtual channels as stream data
Determine network conditions based on transmission of the stream data
Set quality of the video item to be transmitted as the livestream based on the network conditions
END

FIG. 1

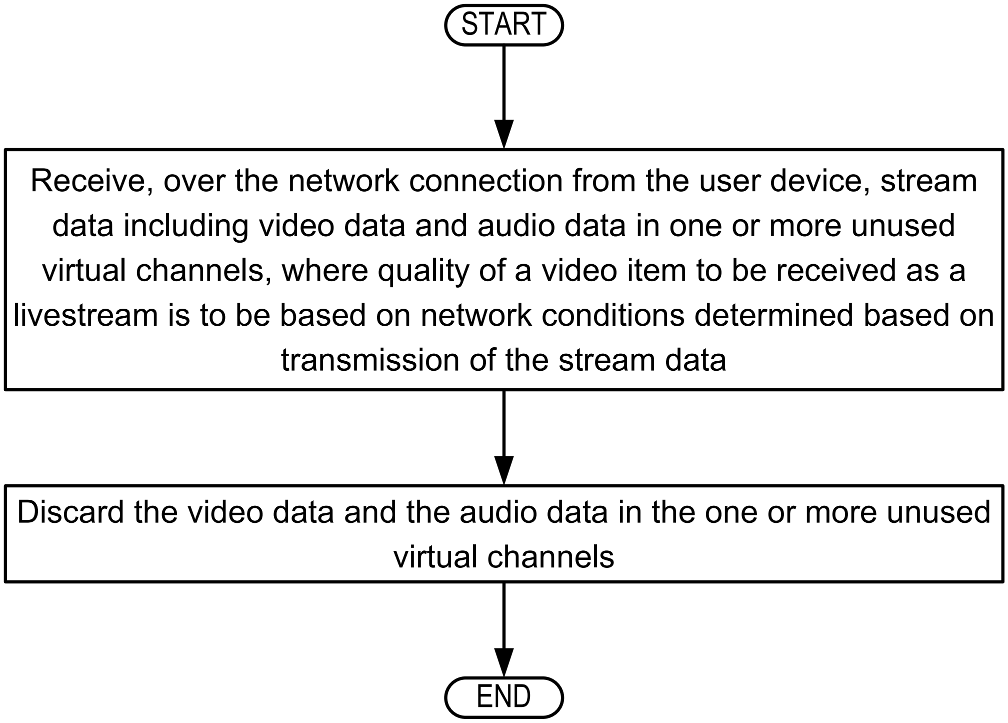
START
Receive, over the network connection from the user device, stream data including video data and audio data in one or more unused virtual channels, where quality of a video item to be received as a livestream is to be based on network conditions determined based on transmission of the stream data
Discard the video data and the audio data in the one or more unused virtual channels
END

FIG. 2