June 2022

MEASURING END-TO-END MEDIA LATENCY IN VIDEOCONFERENCING USING AUDIO WATERMARKING

Ole Gauteplass

Ragnvald Barth

MEASURING END-TO-END MEDIA LATENCY IN VIDEOCONFERENCING USING AUDIO WATERMARKING

AUTHORS:
Ole Gauteplass
Ragnvald Barth

ABSTRACT

An important factor for ensuring high quality collaboration meetings is low end-to-end audio and video latency between the meeting participants. Traditional network tools measure network latency only, and not the additional latency that is introduced by media processing. Measuring the true media latency between participants in a collaboration meeting is thus important. Preferably, such measurements are made during every call using signals that are undetectable by a user. Techniques are presented herein that support the encoding of an inaudible watermark message into the audio streams of a video conference. By measuring the roundtrip time of such a message, the measurement of end-to-end media latency becomes possible. Such measurements may be used to create reports for system administrators, containing statistics regarding the media latency in their collaboration network, that may be used to identify the parts of the network that have unacceptably high media latency.

DETAILED DESCRIPTION

An important factor for ensuring high quality collaboration meetings is low end-to-end audio and video latency between the meeting participants. Measuring the true media latency between participants in a collaboration meeting is thus important. Such measurements may be used to create reports for system administrators containing statistics regarding media latency in their collaboration network. That information may be used to identify the parts of the network that have unacceptably high media latency. It is advantageous that the above-described measurements be made during every call with signals that are undetectable by a user.

Using the media data itself to measure media latency ensures that the measurement reflects the actual media latency. Traditional network tools such as `ping` and `iperf` measure network latency only, and not the additional latency that is introduced by media processing.

Some existing solutions develop the type of measurements that were described above through audible test signals in separate test calls.

To address the types of challenges that were described above, techniques are presented herein that support the use of inaudible signals so that the measurements may be made during a normal conference call. Using the media data makes the solution independent of backend and network protocols. For example, an endpoint does not need to know if media is sent directly to another endpoint through the Real-time Transport Protocol (RTP) Control Protocol (RTCP) or if it is sent through a transcoding intermediary that resides in the cloud.

Aspects of the techniques presented herein encompass three main steps – first, sending a request message; second, responding to a received request message with a response message; and third, receiving a response message – each of which will be described below.

Under the first step (which encompasses the sending of a request message), a collaboration endpoint or application (hereafter denoted as endpoint A) may initiate a latency measurement by encoding a data request message as an inaudible audio watermark in the audio stream before transmission onto the network. Such a request message contains a unique identifier for endpoint A (such as, for example, the endpoint's media access control (MAC) address).

During the second step (which encompasses responding to a request message) the request message (as described above) is received by another participant (hereafter denoted as endpoint B) and decoded. Endpoint B may wait for an arbitrary amount of time before responding to the request message. Endpoint B encodes a response message as an audio watermark and transmits it to the network. The response message contains a unique identifier for endpoint B, the unique identifier for endpoint A (that was received in the request message), and the duration of the interval between receiving the request message and sending the response message minus endpoint B's local input/output (I/O) delay.

Alternatively, endpoint B may elect to wait for the same amount of time as its I/O delay before sending the response message. In such a case there is no need to include time information in the response message since it will be zero.

Under the third step (which encompasses receiving a response message), endpoint A receives and then decodes the response message (as described above) that was received from endpoint B. Endpoint A may then verify that same is in fact a response message from endpoint B to endpoint A by inspecting the unique identifiers.

Figure 1, below, depicts elements of the exchanges that were described above.

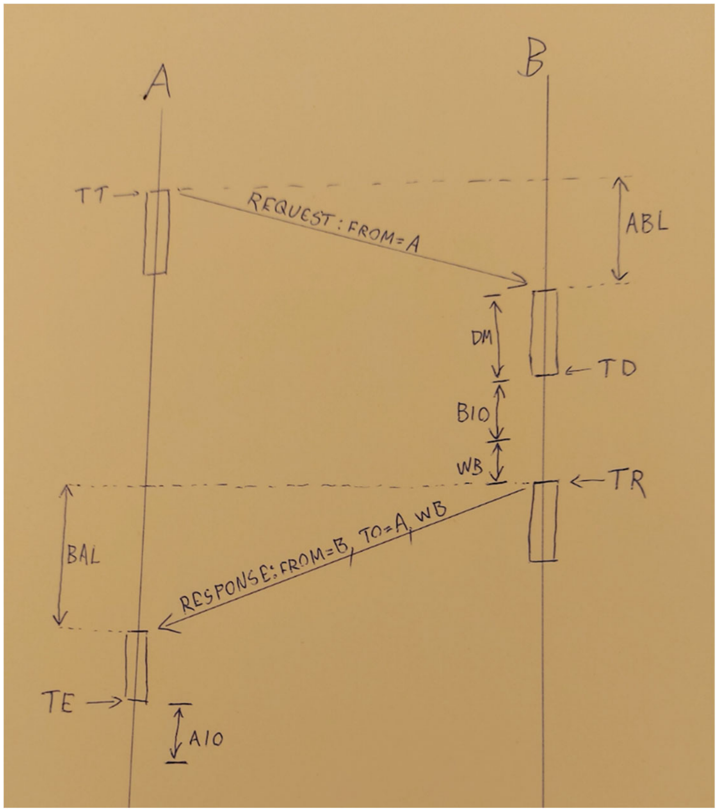

Figure 1: Exemplary Exchanges

Figure 1, above, identifies a number of individual measures which, according to aspects of the techniques presented herein, may be used to calculate end-to-end media latency. The individual measures are defined in Table 1, below.

Table 1: Individual Measures

Measure	Description
TT	Transmit time. Timestamp measured by endpoint A for the start of the request message.
DM	Message duration.
TD	Decode time. Timestamp measured by endpoint B for when the request message was fully decoded.
BIO	Local audio I/O delay at endpoint B.
TR	Respond time. Timestamp measured by endpoint B for the start of the response message.
TE	End time. Timestamp measured by endpoint A for when the response message was fully decoded.
AIO	Local audio I/O delay at endpoint A.
ABL	Endpoint A to endpoint B latency.
BAL	Endpoint B to endpoint A latency.

Referring to Figure 1 and Table 1, both above, endpoint B may calculate the wait time at endpoint B as WB = TR - TD – BIO. The value of the measure WB may then be included (as depicted in Figure 1) in the response message that is sent from endpoint B to endpoint A.

Referring, again, to Figure 1 and Table 1, both above, the total latency, which may be calculated by endpoint A, is equal to the sum of the measures ABL and BAL which, in turn, is equal to the value of TE - TT - 2*DM - WB + AIO.

Aspects of the techniques presented herein provide support for collision avoidance. If the audio streams from multiple participants are mixed, each stream containing a watermark message, collisions may occur. Such collisions may, among other things, disturb the decoding process. A random delay between receiving a request message and sending a response may be employed to reduce the risk of collisions. Alternatively, where an arbitrary delay is not allowed, collisions may be avoided by letting endpoint B randomly decide whether to respond at all. Additionally, a random delay before sending a request message may also be selected.

Further aspects of the techniques presented herein encompass support for spread spectrum techniques. An audio watermark (as described above) may be encoded before or after the lossy audio compression in the transmitting endpoint and decoded before or after lossy audio decompression in the receiving endpoint. Importantly, the watermark should be resilient to lossy transcoding in the cloud service.

An audio watermark data transmission, as described above, is in effect a low bitrate data channel from one endpoint to another endpoint and then back again. A transmission does not need to take place very often, and the bitrate may be very low. Any loss of data is not critical, as an endpoint can simply retry after a timeout period. The requirement for accuracy and precision in the above measurements is on the order of 10 milliseconds (ms). It should be noted that the method that was described above does not depend upon time being synchronized between two endpoints. Spread spectrum techniques are well suited for this purpose, as it makes it possible to hide the transmission below the noise floor and to accurately determine the timing of a decoded message.

Still further aspects of the techniques presented herein encompass support for various reporting capabilities. For example, the measurements (as described and illustrated above) may be collected by an endpoint and then uploaded to a cloud management service to create reports for system administrators that capture indications of media latency in their collaboration network. Such information may be used to identify the parts of a network that have unacceptably high media latency.

For simplicity of exposition, elements of the above narrative employed an example comprising two participants (endpoints A and B). It is important to note that aspects of the techniques presented herein encompass support for environments containing multiple participants.

As just one example, consider three parties (which may be identified as Party A, Party B, and Party C) that are connected in a collaboration conference.

After a random delay, one of the parties (e.g., Party A) will send a request message. Parties B and C will both receive the request message and will both wait for a random period of time before responding. Assume that Party C responds first with a response message:

```
From=C   To=A   WC=wait time at C
```

Party A receives the response message from Party C and then calculates and reports the Party A-to-Party C roundtrip time.

Next, Party B sends a response message:

```
From=B   To=A   WB=wait time at B
```

Party A receives the response message from Party B and then calculates and reports the Party A-to-Party B roundtrip time.

Thereafter, following a random delay, one of the Parties A, B, or C will start the measurement process over again by sending a request message that the other two participants may collect and may respond to. Through such a process each of the three participants will keep measuring the roundtrip time to the other participants.

It is important to note that, during a collaboration conference a media intermediary in the cloud may decide to only forward audio from the loudest speaker(s). The request and response messages (as described and illustrated above) will be lost if the audio is not transmitted. Media latency, and thus the measuring of media latency, is far more important when audio flows in both directions.

In summary, techniques have been presented herein that support the encoding of an inaudible watermark message into the audio streams of a video conference. By measuring the roundtrip time of such a message, the measurement of end-to-end media latency becomes possible. Such measurements may be used to create reports for system administrators, containing statistics regarding the media latency in their collaboration network, that may be used to identify the parts of the network that have unacceptably high media latency.