Technical Disclosure Commons

Defensive Publications Series

October 17, 2018

ZERO SECOND MEETING JOINS

Jonathan Rosenberg

ZERO SECOND MEETING JOINS

AUTHORS:
Jonathan Rosenberg

ABSTRACT

Techniques are described herein for causing a client to pre-join a meeting while using signaling to indicate that it is “invisible” and should not show up in the roster. Furthermore, it does not render any incoming Audio/Video (AV), and does not send any from the camera or microphone. When the user joins, the client starts rendering the media it has already been receiving and then starts sending camera and microphone content. This makes the join process more like an “unmute” operation, which is entirely local and thus can happen instantly.

DETAILED DESCRIPTION

The amount of time required to join a meeting is a key problem today. It can take anywhere from a few seconds to tens of seconds for a user to join a meeting. Ideally this time would be reduced to zero (e.g., under 500 ms).

The starting assumption is that a meeting has been scheduled in a calendar, and as a consequence of integrations with calendaring systems, the meeting is joinable by a join button. This means that the end user client or TelePresence (TP) endpoint has a join button which manifests at the time of the meeting. It further assumes that the meeting system has obtained the meeting Identifier (ID) and other meeting metadata from the calendar invitation.

When a client has a join button to join a meeting and is in the foreground, at the time of the start of the meeting, the client or device does a “pre-join”. It joins the meeting as it normally would if the user had pressed join. For a Personal Computer (PC) client, the join may occur using Voice Over Internet Protocol (VOIP) and not a Public Switched Telephone Network (PSTN) callback. However, the signaling from the client to the cloud includes an indication that this is a “pre-join”. This signals to the cloud that this participant should not be shown in the meeting roster. The join flow otherwise completes largely as normal, with the following exceptions. First, the client does not actually emit any audio content or live video content. Keep-alive traffic (e.g., silence in the audio codec) is sent to

keep the outbound media path alive. However, no actual camera or microphone content is sent. The keep-alive traffic is marked in some way (using a unique Real-time Transport Protocol (RTP) Payload Type (PT) or frame markers) to indicate that this is pre-join traffic. That prevents the cloud audio and video bridges from forwarding the traffic to any recipients. Second, though the client is receiving live video and audio, it is not rendering them to the screen. It is performing local decodes, however, and just discarding the results. Finally, the roster that is sent to all other participants includes the user or TP device name. However, the roster entry includes a flag that the user is "invisible" and consequently should not be rendered into the roster until they complete the join process.

When the user actually selects the join button, the TP endpoint or client completes the following operations. First, it signals to the cloud that it has completed the join. This will update the roster in the cloud, changing the state for that user from pre-joined/invisible to actually joined. Second, the TP endpoint or client begins rendering locally received audio or video content. Third, it begins sending actual microphone and camera input instead of the silence or keep-alive traffic.

Furthermore, the system allows for audio and video streams to be correlated with users in the roster. This is common practice today using Synchronization Source (SSRC) ID and other RTP header extensions which are signaled to clients in the roster updates. This is important because it is very likely that, once a user hits the join button, their audio and video will reach remote participants in the meeting before the roster update has propagated. It is desirable for remote participants to see the user in the roster at the time their audio/video arrives. To provide this capability, all participants look for inbound audio or video that is associated with an invisible/pre-joined participant. If such AV is received, that participant now considers the user to be visible and fully joined, and renders them in the roster. As such, the invisible users are propagated in the roster to remote participants so that, when media arrives, they can immediately show the user in the roster. This feature is not possible with standard Session Initiation Protocol (SIP)-based meeting joins.

The system may also include a timer in case the user never presses the join button. If, after the timer expires, the user has not joined, the client can exit the meeting. In that case, if the user presses the join button a normal join occurs and the user does not get the

benefit of the zero second join. There is a tradeoff between bandwidth consumption and user experience for the value of this timer.

If the user has a join button for many meetings, the client can pre-join all of them in this way. Optimizations are also possible wherein the client would only join one of them based on the most likely meeting for the end user to join.

In summary, techniques are described herein for causing a client to pre-join a meeting while using signaling to indicate that it is “invisible” and should not show up in the roster. Furthermore, it does not render any incoming Audio/Video (AV), and does not send any from the camera or microphone. When the user joins, the client starts rendering the media it has already been receiving and then starts sending camera and microphone content. This makes the join process more like an “unmute” operation, which is entirely local and thus can happen instantly.