January 2021

Seamless Transfer of Ongoing Video Conference by Touching Devices

Diego Rivas

Sweyn Venderbush

Chris Igo

Seamless Transfer of Ongoing Video Conference by Touching Devices

ABSTRACT

When a user wishes to switch to another device in the middle of a video call, it requires the user to join the already-in-progress call from the other device and subsequently end the call on the previous device. Such an approach is time-consuming, burdensome, and can confuse others in the video call. This disclosure describes techniques to overcome these shortcomings by enabling users to transfer an in-progress video call from one device to another by touching the two devices to each other. User permission is obtained to access call information, which is provided to the target device via a hardware mechanism. An application programming interface (API) is utilized to initiate the video call on the target device and simultaneously terminate it on the source device, thus providing seamless call transfer.

KEYWORDS

- Video call
- Video conference
- Video session
- Conference call
- Call transfer
- Session transfer
- Proximity sensor

BACKGROUND

Nowadays, people often use several different devices for one-to-one or one-to-many video calls for personal as well as business purposes. For work purposes, video conferencing is especially important for telecommuting.

Sometimes a person wishes to switch to another device in the middle of a video call. For instance, someone who joins a video call via a smartphone may later wish to continue the call on a desktop computer or tablet device. Such switches may be driven by a variety of reasons, such as differences in device capabilities and bandwidth, battery level, change of location, changes in external circumstances, etc.

When a switch is desired, the typical solution is to join the already-in-progress call from another device (using the call dial-in information or URL) and subsequently end the call on the previous device. With such an approach, there is a short period of time during which both devices are connected to the call. Although workable, the approach has several shortcomings. In particular, it is time-consuming, imposes burden of coordinating actions between the two devices, and can confuse others in the call during the period when a person that is in the process of switching is connected to the call with two devices. Moreover, having two devices in close proximity connected simultaneously to the same video call can create screeching or echo due to audio feedback if one of the device microphones is not muted.

These shortcomings result in a burdensome and frustrating user experience (UX) when switching devices during a video call, leading to lost productivity. The adoption of video calling for everyday personal and work purposes coupled with ownership of multiple devices with video conferencing functionality increases the negative impact of these inefficiencies.

DESCRIPTION

Interactions involving the use of two devices can leverage the use of technologies, such as RFID tags, Bluetooth, ultra-wide band, near field communications (NFC) etc., for device identification and proximity detection. Device proximity and/or tapping have been used as a means to perform various operations, such as file transfers, contactless payments, etc. However, the operations supported using such interactions do not currently include video calls.

This disclosure describes techniques for users to transfer an in-progress video call from one device to another by touching the two devices to each other. When the user touches the two devices, the ongoing video call on one device is seamlessly switched to the other device in real time. The switch is performed by transferring the video calling session active on the first device instead of terminating the session and creating a new session by joining the call separately from the other device. The transfer operation is performed fast enough that the user switching devices as well as other parties in the video call do not perceive any changes because of the device switch. The seamless touch-based call transfer between two devices is achieved via a combination of hardware and software components.

The hardware component enables the mechanisms for device identification and proximity detection, with user permission. Such components can be sub-components within the devices themselves. For instance, such subcomponents can include RFID tags, magnetic sensors, low power cameras, etc. When the user touches two devices equipped with the necessary hardware components, the device with the ongoing video call can generate and provide a signal that triggers an electrical switch on the other device (to which the call is to be transferred). If the user permits, triggering the electrical switch results in the initiation of appropriate logic in the other device to receive the video call session transferred from the first device.

Once the call transfer initiation logic is triggered by the hardware components, the transfer process is achieved via software components that can be distributed across multiple stack layers. If the user permits, the software components at the lower layers of the stack pass the logic from the hardware circuits to the requisite Application Programming Interface (API). Subsequently, the API is employed to generate appropriate function calls (e.g., to the specific video calling application) to indicate to the first device that the in-progress video session is to be transferred and to the second device that a video session is about to be spawned as a result of an incoming session transfer from another device.

The API calls facilitate information exchange with software components at the higher, application layers of the stack. The information exchange facilitates identification of specific video call sessions using relevant metadata, such as a session identifier. For instance, with user permission, an API query to the application layer can be performed on the first device to obtain information indicating the video call to be transferred and passed along to the second device.

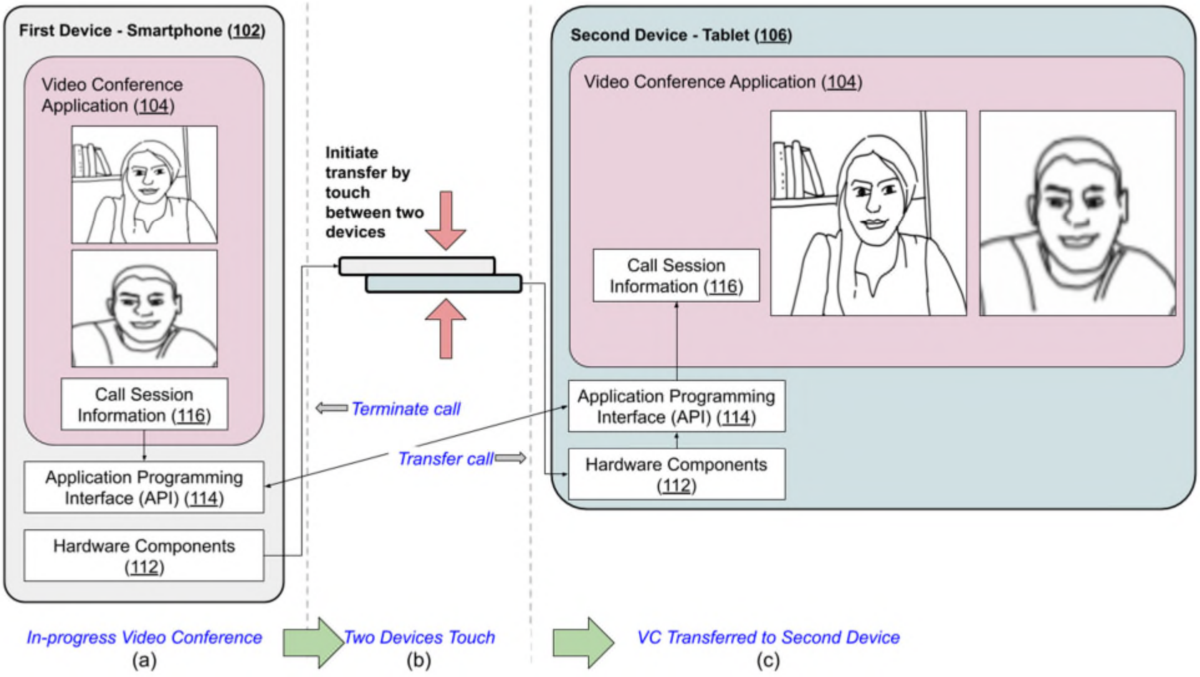
First Device - Smartphone (102)
Video Conference Application (104)
Call Session Information (116)
Application Programming Interface (API) (114)
Hardware Components (112)
Initiate transfer by touch between two devices
Terminate call
Transfer call
Second Device - Tablet (106)
Video Conference Application (104)
Call Session Information (116)
Application Programming Interface (API) (114)
Hardware Components (112)
In-progress Video Conference
(a)
Two Devices Touch
(b)
VC Transferred to Second Device
(c)

Fig. 1: Seamless transfer of ongoing video conference by touching devices

Fig. 1 shows an example of operational implementation of the techniques described in this disclosure. A user wishes to switch an ongoing call (Fig. 1a) being conducted via a video conferencing application (104) on a first user device (102) to the same application on a second user device (106). The user initiates the transfer by touching the two devices to each other as shown in Fig. 1(b).

Hardware components (112) within the two devices recognize that the two devices have touched, triggering logic on the second device to invoke the API (114) to initiate transfer of the ongoing call from the first device to the second. API calls are used to obtain session information (116) for the in-progress call and to pass the information along to the second device. As shown in Fig. 1(c), the session information is used to join the call to the second device, simultaneously terminating it on the first device.

With user permission, it can be confirmed that the same user owns both devices, and in such a case, no additional authentication is required when transferring the call. However, the described techniques can also include authentication capabilities with appropriate additions to the hardware and/or software capabilities, such as the use of unique hardware tags to identify devices. With such a configuration, users are required to confirm device ownership via appropriate authentication and approve the target device for video call transfer via touch interaction.

Video call transfer using device touch can also be implemented using only software components, coupled with various proximity sensors and/or Bluetooth. However, it is possible that operation using such an approach is unreliable since it requires fine-tuned coordination between the software components at multiple layers and the sensors. In contrast, the use of hardware components as described herein provide a seamless and reliable end-to-end solution for transferring video calls using the intuitive action of touching two devices. Notably, implementing the described techniques can be achieved simply by planning the architecture for end-to-end support for touch-based call transfer, thus making it possible to achieve the functionality without significant hardware and/or software changes.

Implementation of the techniques described in this disclosure replaces multiple cumbersome actions for transferring video calls with an efficient single-touch operation. The operation can be particularly useful in situations in which a quick call transfer across devices is essential, such as running out of device battery during an ongoing call or having to move from a desktop computer to a mobile device during the call. The techniques can be implemented to support any device, operating system, or application that provide video and/or audio calling capabilities and can handle call transfers for any type of calling mode (i.e., one-to-one, one-to-many, many-to-many). Implementation of the described techniques, e.g., as part of an operating system, can enhance the user experience (UX) of video calling and conferencing for personal and business purposes.

Further to the descriptions above, a user is provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's ongoing calls, session identifiers, a user's devices, or a user's preferences), and if the user is sent content or communications from a server. In addition, certain data is treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity is treated so that no personally identifiable information can be determined for the user. Thus, the user has control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes techniques to overcome these shortcomings by enabling users to transfer an in-progress video call from one device to another by touching the two devices to each other. User permission is obtained to access call information, which is provided to the target device via a hardware mechanism. An application programming interface (API) is utilized to initiate the video call on the target device and simultaneously terminate it on the source device, thus providing seamless call transfer.