# Handoff with DSP Support: Enabling Seamless Voice Communications across Heterogeneous Telephony Systems on Dual-Mode Mobile Devices

Hung-Yun Hsieh, Chung-Wei Li, and Hsiao-Pu Lin

**Abstract**—Many mobile handsets manufactured today are equipped with wireless data modules that can provide users with the alternative to access Internet telephony service for voice communications. Due to the limited coverage area of wireless data service, however, a call established through VoIP service may need to be transferred transparently to mobile telephony service, and vice versa, for maintaining voice call continuity. In this paper, we investigate the problem of supporting seamless voice communications across heterogeneous telephony systems on dual-mode mobile devices such as GSM-Wi-Fi handsets. While related work has investigated the problem of seamless vertical handoffs across heterogeneous wireless data networks, solutions based on packet-switched protocols cannot be used in this context since GSM is a circuit-switched telephony system. Rather, to enable seamless voice communications across heterogeneous telephony systems, we show in this paper that the support of digital signal processing (DSP) techniques during handoffs is critical. To substantiate our argument, we start with a framework based on the Session Initiation Protocol (SIP) for supporting vertical handoffs on dual-mode mobile devices. We then identify the key obstacle in achieving seamless voice call continuity across circuit-switched and packet-switched systems and explain why a "make-before-break" handoff with DSP support is necessary. We thus propose a solution that incorporates time scaling algorithms to process voice streams during handoffs for supporting seamless voice call continuity, and we investigate mechanisms to reduce the overheads of the proposed solution. To evaluate the performance of the proposed solution, we conduct testbed experiments using a GSM-Wi-Fi dual-mode notebook. Evaluation results show that such a cross-disciplinary solution involving signal processing and networking can effectively support seamless voice communications across heterogeneous telephony systems.

**Index Terms**—Heterogeneous wireless networks, multihomed mobile device, VoIP, SIP, WSOLA, PESQ.

## 1 Introduction

Voice over IP (VoIP) is one of the fastest growing technologies in the Internet today. Using advanced voice compression techniques and efficient real-time transport mechanisms, the VoIP technology allows phone calls to be made over packet-switched networks without relying on dedicated circuit-switched infrastructure required by conventional telephony systems. Since the VoIP technology operates over packet-switched networks, it can leverage the richness of Internet services to create new services combining data, voice, and other multimedia applications. More importantly, it can ride on the pervasiveness of Internet infrastructure to extend the reach of telephony service wherever Internet is accessible.

As the Internet becomes increasingly wireless, therefore, so does VoIP. In particular, voice over WLAN (VoWLAN) has attracted a lot of attention due to the prevalence of the IEEE 802.11-based WLAN technology (Wi-Fi). Compared with other wireless access technologies, WLANs can be deployed rather quickly and offer much higher data rate with substantially lower cost. Placing VoIP calls from inside these Wi-Fi hot spots hence enables untethered voice communications without incurring the cost required by existing mobile cellular telephony systems such as GSM, PCS, IS-95, and 3G.[1]

Despite the potential benefits of the VoWLAN technology, the fact that WLANs typically have only spotty coverage limits the applicability of VoWLAN-enabled handsets. A VoWLAN call can be placed or received only when the user is within the reach of WLANs. When moving outside the coverage of Wi-Fi hot spots, for example, a mobile user may need to resort to his/her GSM handset for getting connected. While more and more cities worldwide have undertaken plans to build large-scale Wi-Fi networks or even WiMax networks with the goal of enabling citywide wireless Internet access, the coverage still cannot parallel that provided by existing cellular telephony systems, especially in suburban or rural areas. It is conceivable, therefore, that a traveling VoWLAN user may also need to bring a GSM handset for assurance of voice communications.

To take advantage of the *opportunistic* nature of the VoWLAN service without the need to bring two handsets, the concept of dual-mode handsets has emerged and come to the limelight. Many GSM-Wi-Fi, EDGE-Wi-Fi, 3G-Wi-Fi, and HSDPA-Wi-Fi dual-mode handsets and PDA phones have made their debut recently [1], [2], [3]. Such dual-mode handsets by design can be used as either VoWLAN or GSM

• *The authors are with the Graduate Institute of Communication Engineering and the Department of Electrical Engineering, College of Electrical Engineering and Computer Science, National Taiwan University, Room 546, EE-II Building, No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan 106, R.O.C. E-mail: hyhsieh@cc.ee.ntu.edu.tw.*

*Manuscript received 8 Sept. 2006; revised 21 Aug. 2007; accepted 12 May 2008; published online 4 June 2008.*
*For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0231-0906.*
*Digital Object Identifier no. 10.1109/TMC.2008.87.*

1. We use GSM as a general term to represent variants of public mobile cellular telephony systems hereafter.

handsets at different times and places depending on service availability and user preference. However, a key functionality that truly differentiates one dual-mode handset from *two* single-mode handsets is the ability to allow *seamless migration of a phone call from one telephony system to another*.[2] Take the GSM-Wi-Fi dual-mode handset for example. A call connected through the Wi-Fi mode (VoWLAN call) should be migrated to the GSM mode (GSM call) when the user moves beyond the reach of Wi-Fi service. If handoffs can take place automatically (subject to user preference) without disrupting ongoing voice communications, the benefits of dual-mode handsets can be truly exploited.

Related work has investigated the problem of vertical handoffs across heterogeneous wireless overlay networks using techniques such as packet buffering and forwarding, double casting, path rerouting, and packet striping for achieving fast and seamless handoffs [4], [5], [6], [7]. These solutions, however, cannot be applied to solve the problem faced by dual-mode handsets. GSM, like other mobile cellular and public switched telephone networks, is a *circuit-switched telephony system* that establishes the call using dedicated "circuits" without following the *store-and-forward* paradigm in packet-switched data networks. In circuit-switching networks, each circuit that is dedicated to one call in progress cannot be used by other callers until the circuit is released. Therefore, while it is possible to multiplex packets from different wireless data networks (e.g., through packet buffering and forwarding at the base station) for ensuring seamless migration of data connections, it is nontrivial to multiplex one circuit-switched call with one packet-switched call using only packet-based techniques. Moreover, circuit-switched networks have adopted the "*smart network, dumb terminal*" paradigm opposite to that in the Internet. Hence, it is difficult, if not impossible, to control circuit-switched voice effectively on end devices as can be done in packet-switched networks (e.g., through the manipulation of the packet header for rerouting). While it is possible to leverage infrastructure support for interworking the two telephony systems and performing conversion between circuit-switched and packet-switched voice inside the network [8], [9], [10], such a solution might not always be desirable for the dual-mode user due to the *extra cost and overhead* incurred by directing VoWLAN traffic to the GSM core network. We discuss in Section 6 the merits and shortcomings of infrastructure-based approaches.

In this paper, we investigate the problem of supporting seamless vertical handoffs across heterogeneous telephony systems on dual-mode mobile devices including GSM-Wi-Fi dual-mode handsets, PDAs, and other portable devices equipped with both wireless NICs and modems. The scenario we consider involves voice communications between a dual-mode mobile device and a remote VoIP client. The remote VoIP client can be another dual-mode mobile device, a wired or wireless softphone, or a VoIP gateway installed in the enterprise, campus, or residential building. The goal is to allow the dual-mode device to switch transparently from Wi-Fi to GSM modes, and vice versa, during a phone call with the remote peer. We target a solution that can be implemented on the end devices without requiring new network entities for handoff operations to be deployed. While it is not the scope of this paper to argue for or against infrastructure-based approaches, motivated by the high mobile handset replacement rate and the fact that more and more sophisticated functionalities are being added to the handsets, we believe that future wireless networks can benefit from more intelligent mobile handsets for services such as vertical handoffs. In this way, not only can infrastructure be alleviated of the overheads, but end devices can control whether the seamless handoff functionality should be activated depending on user preference and the peer and the type of communications. Clearly, new functionalities added to dual-mode handsets do not preclude adoption of infrastructure-based solutions in the future.

Toward this goal, we start by proposing a vertical handoff framework across heterogeneous telephony systems based on the Session Initiation Protocol (SIP), the standards developed by IETF for initiating, modifying, and terminating multimedia sessions including voice communications [11]. SIP is an end-to-end protocol that works independently of the type of session that is being established, and it has been designed to interwork transparently with PSTN (public switched telephone network) signaling including SS7 ISUP (ISDN user part) [12]. We first investigate different approaches based on SIP that have been proposed in different contexts for supporting mobility, including using RE-INVITE for midcall mobility [13] and REFER for call transfer [14]. We then identify the problems of these approaches when used directly in the target environment and find that a desirable solution should allow for "make-before-break" soft handoffs due to the problem of *prolonged call setup time* across heterogeneous telephony systems. We thus propose the framework for vertical handoffs on dual-mode mobile devices that allows the existing call to be terminated gracefully after the new call has been established.

While the proposed framework allows for session continuity during handoffs, we have found that during the period when the two calls coexist, the voice streams of the two calls cannot be simply mixed due to *latency mismatch between the paths traversed by the two calls, respectively*. As we discuss in Section 3, the requirement for synchronization between GSM and VoWLAN calls is different from that in conventional multiparty conferences [15] for reasons including the *echo effect* and the *protocol heterogeneity* of the two calls. To address the problem, we investigate digital signal processing (DSP) techniques that operate on audio samples for ensuring seamless mixing of the two voice streams. The advantage of using DSP techniques is that it does not matter whether the voice stream comes from circuit-switched or packet-switched calls. In the proposed solution, the handoff process first uses a time alignment algorithm to identify where the two voice streams should be "glued" and then uses a time scaling algorithm to modify speech for *closing the audio gap* between the two voice streams that results from latency mismatch of the paths.

Compared to an approach that cuts off the old voice stream and switches immediately to the new one without any processing, the proposed solution can potentially incur additional power consumption and processing overheads since it requires the two voice streams to overlap in time for

2. In such dual-mode handsets, each mode may still involve a combination of wireless technologies in different frequency bands (e.g., GSM 900/1800/1900 and WCDMA/CDMA2000 for the GSM mode and 802.11 a/b/g for the Wi-Fi mode), but a distinguishing feature from conventional "multiband" handsets is the ability to support voice communications *simultaneously* through *heterogeneous* telephony technologies involving circuit-switched cellular telephony and packet-switched Internet telephony.

signal processing during the handoff period. To reduce the overheads of the proposed solution, therefore, the overlapping time of the two voice streams and the computational complexity of the DSP algorithms need to be maintained as low as possible. We thus investigate along the design space of the DSP algorithms for reducing the overlapping time and the computational complexity without significant degradation in the accuracy of the algorithms and, hence, the quality of speech during handoff. To evaluate the performance of the proposed solution, we setup a SIP testbed and use a dual-mode notebook with GSM and Wi-Fi connectivity for voice communication with a remote SIP phone. We record the voice stream received on the dual-mode notebook when it switches from Wi-Fi to GSM with and without the proposed solution during the handoff period. A survey with 30 users on the voice quality of the recorded voice streams shows that the proposed solution does provide a more pleasing call transfer experience across heterogeneous telephony systems. While it is not the scope of this paper to study various optimizations tailored to existing hardware, evaluation results show that such a cross-layer solution involving signal processing and networking sheds a promising direction for further optimization and research in the future.

The rest of this paper is organized as follows: Section 2 presents background knowledge on SIP support for mobility and motivates the need for a solution based on "make-before-break" soft handoffs. Section 3 presents a SIP-based vertical handoff framework, and Section 4 presents the DSP techniques to be used in tandem with the framework to enable seamless handoffs between circuit-switched and packet-switched voice streams. Section 5 presents testbed results. Finally, Section 6 discusses related work and concludes the paper.

## 2 Background and Motivation

In this section, we first present a brief overview of SIP and then discuss related work on SIP support for mobility. Finally, we identify the problems with existing solutions in the target environment and motivate a solution that can support "make-before-break" soft handoffs.

### 2.1 Session Initiation Protocol

The SIP is an application layer signaling protocol developed by IETF for creating, modifying, and terminating sessions with one or more participants [11]. A session in SIP is a collection of participants and the media streams (including audio, video, and text) between them for the purposes of communication. As a signaling protocol, SIP works in concert with other existing Internet protocols such as Session Description Protocol (SDP), Real-time Transport Protocol (RTP), and Real-time Control Protocol (RTCP) for control and delivery of multimedia data. While SIP works independently of the underlying transport protocols and the type of session that is being established, it has been designed to support traditional PSTN telephony services such as call forwarding, call transfer, and three-way conferencing.

SIP is based on an HTTP-like request/response transaction model, where the SIP end system (user agent or UA) generating requests is called a user agent client, and the one responding to requests is called a user agent server. A SIP transaction consists of a request from the client to invoke a particular method (function) on the server and at least one response triggered by the request. SIP is an end-to-end protocol that does not rely on any infrastructure-based servers to create, modify, or terminate sessions between end systems as long as requests and responses can be delivered to target user agents. To facilitate directory lookup and location service, however, several SIP servers such as registration servers, redirect servers, and proxy servers are often introduced in SIP networks. It has to be noted that while these servers may help forward or respond to requests before the session is setup, they are not required once the end systems learn of the addresses of remote entities. Such a design along with the fact that the intelligence in a SIP network is located in end systems allows *new services to be created and deployed by changing only end systems without any changes in the SIP network*. This is very different from the signaling system in the PSTN where the intelligence lies in the core of the network and new services cannot be easily supported without upgrading the entire infrastructure.

To allow phone calls to be made between SIP phones and conventional PSTN phones, SIP providers have introduced gateways to interwork the two networks. A gateway works by acting as a *user agent* for the SIP network and a *terminating switch* for the PSTN. It typically consists of three components (collocated or distributed): signaling gateway for receiving signaling on the PSTN side and encapsulating it over IP for routing (and vice versa), media gateway for terminating a PCM trunk on the PSTN side and bridging it to packetized bitstreams through RTP payloads (and vice versa), and media gateway controller for controlling the media gateway and converting between the PSTN signaling protocol and the SIP. In addition to bridging calls between PSTN and SIP networks, it is possible to use the SIP network as the backbone network between two PSTN phones. Interested readers are referred to [12] for technical details on the SIP-T framework for PSTN-SIP-PSTN interworking.

### 2.2 SIP Support for Mobility

In a wireless environment, it is possible that the IP address of the participant might change due to mobility (e.g., changes of the point of attachment). To keep the VoIP session after the participant acquires a new IP address, it is necessary to "modify" the session to reflect the new peer relationship and ensure proper multimedia delivery. In SIP, end systems can send an INVITE request within an established session (known as RE-INVITE) for modifying session parameters, including changing addresses or ports, adding media streams, and deleting media streams. Related work has proposed mechanisms to allow midcall mobility (handoffs during a call) by using the RE-INVITE request during the call for modifying the IP address [13].

While SIP mobility based on the RE-INVITE method can tackle the problem of address change during an active session, it cannot be used directly for handoffs between GSM and Wi-Fi modes on a dual-mode mobile device. The reason is that RE-INVITE only allows for change of session parameters between two end systems with *an established session* but handoffs from Wi-Fi to GSM modes require establishment of a new session involving the third user agent—the PSTN gateway (or SIP-GSM gateway)—for connecting the GSM mode and bridging circuit-switched and packet-switched voice streams. The states on the PSTN

TABLE 1
GSM Call to the Remote SIP Client

| Time (sec) | Mean | STD | MAX | MIN |
|---|---|---|---|---|
| Call setup time | 10.266 | 0.577 | 11.619 | 9.475 |
| End-to-end delay | 0.316 | 0.025 | 0.368 | 0.270 |

gateway for the call between the GSM mode and the remote peer need to be established anew; hence, the SIP RE-INVITE method cannot be used "as-is" to solve the target problem.

If we consider the GSM and Wi-Fi modes on the dual-mode device as two independent users that might engage in communications with the remote peer, then handoffs between the two modes can be considered as *call transfer* from one mode to the other. Therefore, another direction toward solving the target problem is through transfer of the two calls. The REFER method indicates that the recipient should contact a third party using the contact information provided in the request [14]. It has been proposed to extend the ability of SIP for supporting enhanced telephony services available in the PSTN, in particular call transfer and third-party call control services. A basic form of call transfer involves the transferor, transfer target, and the transferee. The transferor first provides the transfer target's contact to the transferee using the Refer-To header field, and then terminates the existing session with the transferee after the latter acknowledges the request. The call between the transfer target and the transferee is setup afterwards. It is obvious that the REFER method requires a new session be setup to which the old session is transferred, and hence, it does not suffer from the same problem in the RE-INVITE request, as mentioned before. However, in conventional unattended transfer, the session between the transferor and the transferee is terminated before the session between the transfer target, and the transferee is established. Such a "break-before-make" transfer can be used only when the call setup delay between the transfer target and the transferee is small. We discuss in the following why this approach cannot achieve the desired voice continuity in the target environment.

### 2.3 Motivation for "Make-Before-Break" Handoffs

To understand the problem of "break-before-make" call transfer, we setup a testbed experiment using a dual-mode mobile device and a remote SIP phone (details for testbed setup are presented in Section 5). The dual-mode client can place GSM and VoWLAN calls with the remote SIP client using GSM and Wi-Fi modes, respectively. We measure the characteristics of the paths including call setup time and end-to-end delay between the GSM and Wi-Fi modes and the remote SIP client. The call setup time is measured at the caller (dual-mode client) from the time the call is placed to the time the voice of the callee is heard. It represents the delay for the new call to be fully established during handoff. As we observe in Table 1, for 10 different iterations of experiments, the average call setup time between the GSM mode and the remote SIP client is very long (to the order of 10 seconds). Hence, if "break-before-make" call transfer is used for vertical handoffs from Wi-Fi to GSM modes, the VoWLAN call is terminated before the GSM call is established. In this way, the disruption due to temporal discontinuity in speech would be too long to be unnoticed.

In summary, SIP can be used to setup and release VoWLAN and GSM calls on dual-mode devices. However, existing solutions including session modification based on RE-INVITE and vanilla call transfer based on REFER cannot be directly used for seamless vertical handoffs on dual-mode mobile devices. A desirable solution needs to support "make-before-break" soft handoffs and address the heterogeneity of the two telephony systems used by the dual-mode devices. We present in Sections 3 and 4 the proposed solution for achieving this goal.

## 3 Dual-Mode Vertical Handoff Framework

In this section, we first propose a vertical handoff framework for dual-mode mobile devices that can ensure "make-before-break" soft handoffs and then identify the requirement for DSP support in this framework for achieving seamless vertical handoffs.

### 3.1 Vertical Handoff Framework

The goal of the proposed vertical handoff framework is to ensure "make-before-break" soft handoff while reusing the SIP UA and GSM phone modules on the dual-mode mobile device. Fig. 1 shows the proposed changes to the dual-mode

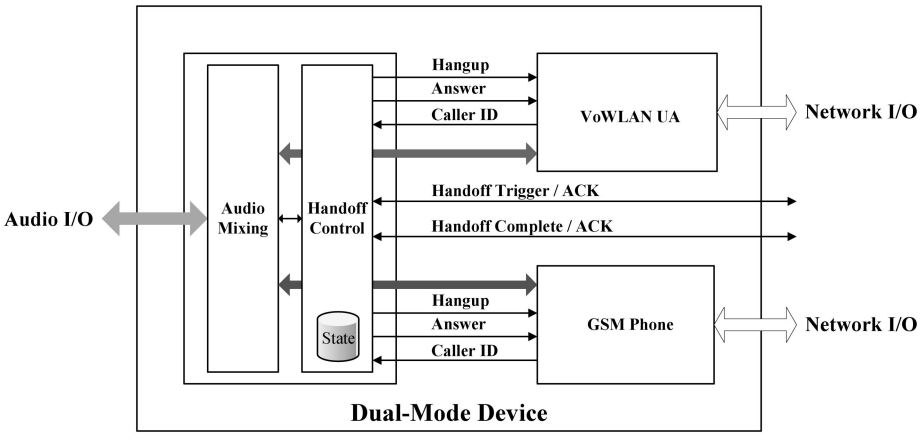


Fig. 1. Dual-mode device functional blocks.

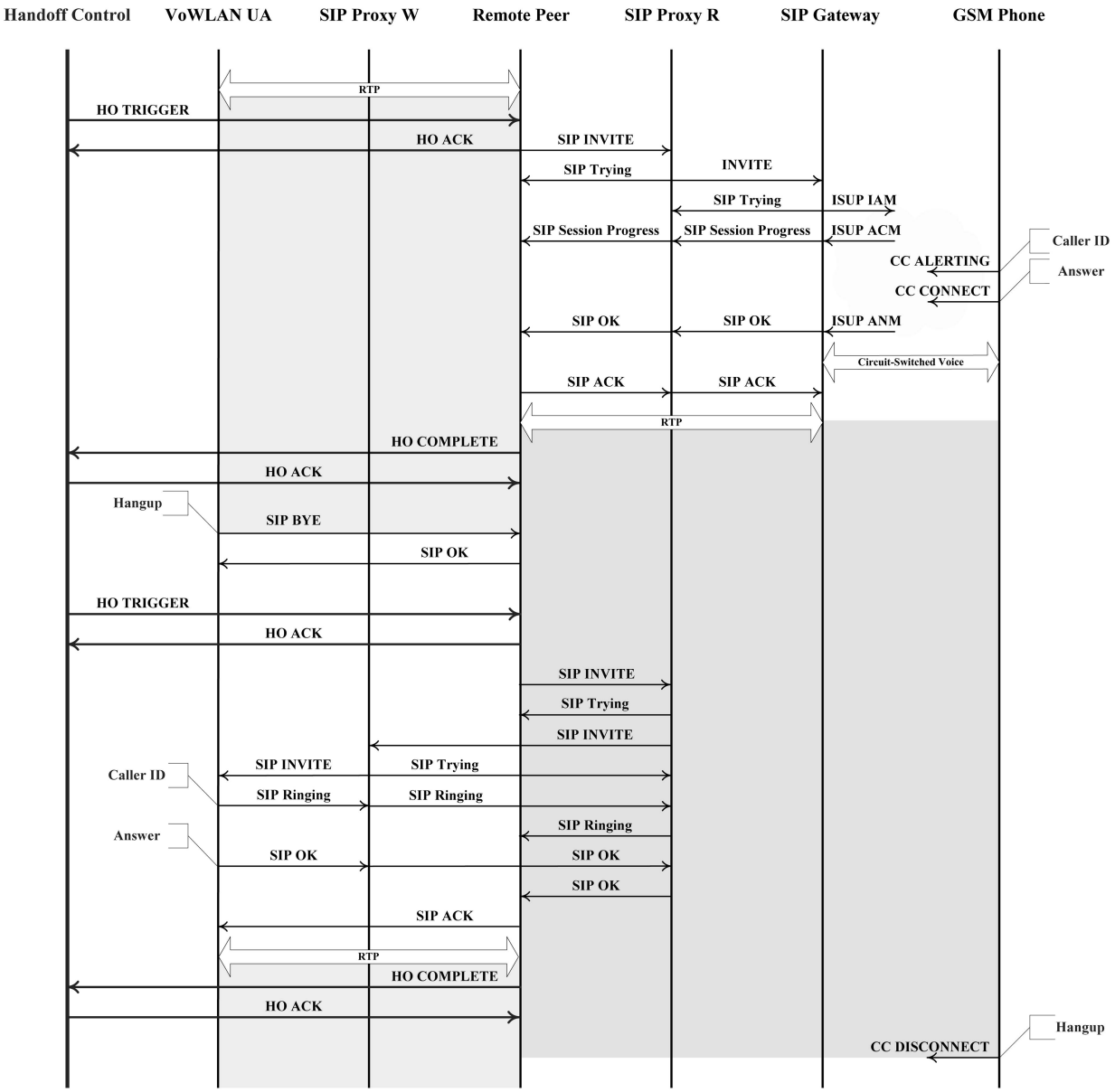


Fig. 2. Vertical handoff between Wi-Fi and GSM.

device for achieving this goal. The *VoWLAN UA* and the *GSM Phone* modules are used by the dual-mode device to make and receive calls through the Wi-Fi and GSM modes, respectively. The *Handoff Control* module controls when handoffs should be triggered, and it keeps track of the handoff progress between the dual-mode device and the remote peer. It also interacts with the *VoWLAN UA* and *GSM Phone* modules to make the latter answer or hang up the call with the remote peer. The *Audio Mixing* module, to be detailed in Section 4, processes voice streams received from the *VoWLAN UA* and *GSM Phone* modules before they are sent to the system device for audio I/O. If the remote peer is not a dual-mode device (i.e., any SIP softphone), then the *Handoff Control* and *Audio Mixing* modules are integrated to the SIP UA for interacting with the two VoIP calls (to/from the Wi-Fi and GSM modes on the dual-mode device) in the control and user planes, respectively.

Fig. 2 shows the handoff process involved for Wi-Fi-to-GSM and GSM-to-Wi-Fi vertical handoffs. Assume that the dual-mode device is initially in communication with the remote peer using the Wi-Fi mode (e.g., in a VoWLAN call). During the call session, the *Handoff Control* module measures the quality of the WLAN link and decides whether a handoff to the GSM mode should be performed. The link quality can be determined based on the received signal strength measurements or RTCP sender/receiver reports on packet loss rates, delays, and jitters. It is also possible to make the handoff decision based on measurements of the speech quality using the chosen objective metrics [16]. If a handoff to the GSM mode is decided, the *Handoff Control* module sends a *Handoff Trigger* command to the remote peer to initiate handoff. *Handoff Trigger* command includes the call state of the VoWLAN call (e.g., the call-ID in the SIP message header that uniquely identifies the call) and the GSM phone number so the remote peer can identify the dual-mode device and match the command to the existing call session. Upon receiving the handoff trigger, the remote peer proceeds to make a call

to the GSM phone number using handshakes as shown in Fig. 2 (some PSTN and GSM signaling messages are omitted due to lack of space).

Once the *GSM Phone* module on the dual-mode device is alerted for the incoming call, the *Caller ID* is exposed to the *Handoff Control* module. Based on its internal state, the *Handoff Control* module matches the caller to the remote peer that the Wi-Fi mode is currently in communication with and then directs the *GSM Phone* module to answer the call using the *Answer* command. After the call between the remote peer and the GSM mode is established and the voice stream is flowing, the remote peer sends a *Handoff Complete* command back to the dual-mode device. Through interaction with the *Audio Mixing* module, the *Handoff Control* module decides when the VoWLAN call should be released. A *Hangup* command from the *Handoff Control* module directs the *VoWLAN UA* to release the call with the remote peer. As shown in Fig. 2, handoff from Wi-Fi to GSM modes is transparent to the user, and the two voice streams can overlap in time without temporal discontinuity (note the shaded areas). Section 4 discusses how the two voice streams are mixed for ensuring seamless handoffs.

The handoff process from GSM to Wi-Fi modes follows similar flows: The *Handoff Control* module first decides whether a handoff to the Wi-Fi mode is necessary via feedback from the WLAN sniffing module. If it is decided that a handoff to the Wi-Fi mode is desirable, the *Handoff Control* module sends the *Handoff Trigger* command (via the Wi-Fi mode) to the remote peer, conveying the SIP URI of the *VoWLAN UA*. The remote peer will proceed to make the VoWLAN call using the SIP INVITE handshakes, as shown in Fig. 2. The *Handoff Control* module can match the caller to the remote peer and answer the call automatically. After the VoWLAN call is setup, the *Handoff Control* module can hang up the GSM call, thus completing GSM-to-Wi-Fi handoff.

We note that while the protocol handshakes shown in Fig. 2 require the remote peer to make the new call (e.g., the GSM call for handoff from Wi-Fi to GSM modes), it is possible to let the dual-mode device make the new call instead. For example, in Wi-Fi-to-GSM handoff, after the *Handoff Control* module receives the acknowledgment to the *Handoff Trigger* command, it may direct the *GSM Phone* module to make a call to the remote peer. The remote peer can be made to answer the GSM call automatically since the GSM phone number has been conveyed to it in the *Handoff Trigger* command. In many GSM billing plans, both the caller and the callee share the cost of communication, while in many others, only the caller is billed for the call made. Therefore, it may matter in some scenarios whether the remote peer or the dual-mode device makes the new call. The *Handoff Trigger* handshake can allow the remote peer and the dual-mode device to negotiate who should make the new call if so desired. Further investigation into the billing issue during vertical handoffs, however, is beyond the scope of this paper.

### 3.2 Requirement for DSP Support

The vertical handoff framework presented in Section 3.1 essentially creates a "three-party conference" among the Wi-Fi mode, GSM mode, and the remote peer. Despite, the two voice streams from Wi-Fi and GSM modes, respectively, cannot be simply mixed for playout as in conventional three-party conferences. For voice conferencing involving dialogs with various participants, related work has shown that synchronization errors less than 120 ms are generally not noticeable [15], [17]. However, since the two voice streams entering the dual-mode device are replicated (subject to transmission degradation), the impact of echo needs to be considered. It is known that the presence of *echo occurs whenever the delay of the replicated signal exceeds 10 ms*, and it becomes apparent to the speaker as reflected voice when the delay exceeds as little as 16 ms [18].

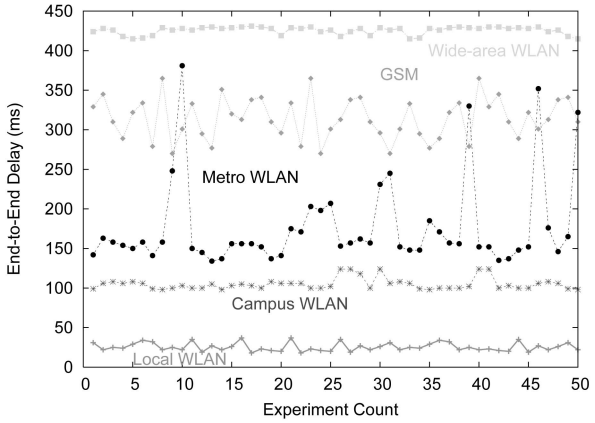


Fig. 3. End-to-end delays of VoWLAN and GSM calls.

To understand the delay mismatch between the two heterogeneous telephony systems on the dual-mode mobile device, we measure the end-to-end delay for VoWLAN and GSM calls between the dual-mode device and the remote SIP client and plot the results in Fig. 3. Different types of WLANs ranging from local WLANs to wide-area mesh WLANs have been tested for making the VoWLAN calls. It is clear that the end-to-end delay for VoWLAN calls varies depending on the type of WLANs the user is connected to, but the delay mismatch between VoWLAN and GSM calls is large enough to create "echo effect" if the two voice streams are simply mixed together on the dual-mode device.

Therefore, the synchronization of the two voice streams during "make-before-break" call transfer is important for achieving seamless handoffs, and the synchronization requirement for audio mixing during vertical handoff is much tighter than that in conventional conferencing applications. Due to the heterogeneity of the two telephony systems involved, existing approaches that leverage information such as RTP time stamp and SSRC for synchronization [15] cannot be used. (Note that the GSM audio is not packetized using conventional RTP packets.) A new approach that can work independently of the network protocols and audio codecs used by the two voice streams thus needs to be designed. We present in the following the DSP techniques to be used in tandem with the proposed vertical handoff framework for achieving this goal.

## 4 DSP for Seamless Vertical Handoffs

In this section, we first present the functional blocks of the *Audio Mixing* module for speech processing on the dual-mode device and an overview of its operations during handoffs. We then present in details the required signal processing operations, including time alignment and time scaling algorithms.

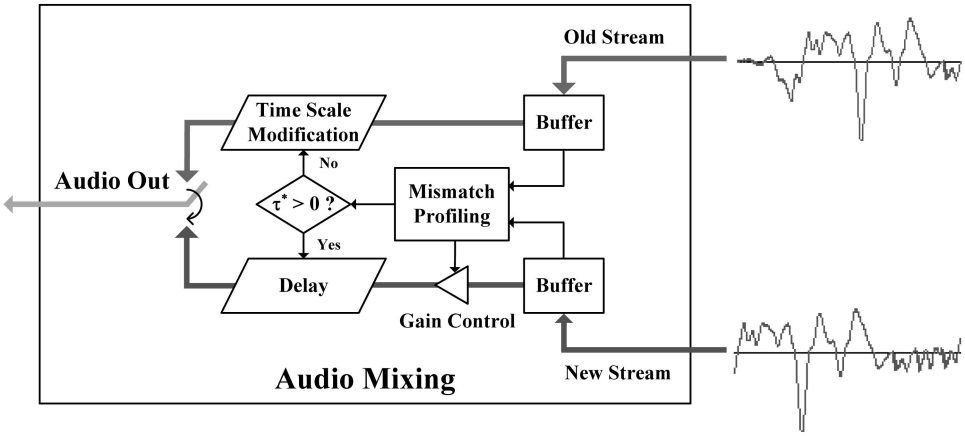


Fig. 4. Audio mixing module.

### 4.1 Overview of Audio Mixing

The functionality of the *Audio Mixing* module introduced in Fig. 1 is to ensure seamless continuation of speech when two voice streams (from GSM and VoWLAN calls) coexist during handoffs. In the proposed architecture, only incoming voice streams (received from the network) are processed by the *Audio Mixing* module using the proposed DSP techniques for output to the earphone. Outgoing voice (received from the microphone) is not processed by the *Audio Mixing* module but sent directly to the *VoWLAN UA* and *GSM Phone* modules for transmissions to the network. Fig. 4 shows the functional blocks of the *Audio Mixing* module.

In the normal operation of the *Audio Mixing* module, incoming voice streams received from either GSM or VoWLAN call are passed directly to the audio output device. After handoff is initiated by the *Handoff Control* module (refer to Fig. 2) and the new voice stream enters the device, the DSP algorithms in the *Audio Mixing* module are activated so the new voice stream can be "glued" seamlessly to the old voice stream. First, since the two incoming voice streams may experience different end-to-end delays after they are sent out by the remote peer, the *Mismatch Profiling* block uses a time alignment algorithm to decide the time relationship of the two voice streams. After the correct time offset $\tau^*$ of the new voice stream (relative to the old one) is identified, the *Gain Control* block adjusts the amplitude of the new voice stream to compensate for the potential gain mismatch of the audio devices used by the *VoWLAN UA* and *GSM Phone* modules.[3]

If it is determined that the new voice stream arrives earlier than the old stream (i.e., the new voice stream experiences a shorter end-to-end delay), the former is delayed in the buffer for proper amount of time before being sent to the audio output device to replace the old voice stream. On the other hand, if the new voice stream experiences a longer end-to-end delay than the old stream, a time scaling algorithm is performed to slow down the playout speed of the old voice stream. Switching from the old voice stream to the new one occurs when the latter catches up with the playout schedule of the former. Through combined operations of time alignment and time scaling, switching from the old voice stream to the new one is free of the echo effect and does not suffer from any temporal discontinuity. After switching, the *Audio Mixing* module notifies the *Handoff Control* module, which then proceeds to hang up the old call with the remote peer, as shown in Fig. 2. In the following, we discuss in details the time alignment and time scaling algorithms performed by the *Audio Mixing* module.

3. We note that while the *VoWLAN UA* module typically uses the default system audio device, on many dual-mode handsets, a separate audio device is used for the *GSM Phone* module.

### 4.2 Time Alignment

Since voice streams received through GSM and VoWLAN calls may experience different end-to-end delays, an abrupt switching from the old stream to the new one will introduce undesirable pop, clip, or repetition in the audio output. Therefore, it is necessary that proper time shift operation is applied to the new voice stream before switching. A time alignment algorithm thus can be used to find the time relationship (e.g., offset of time reference) of the two voice streams. In the time alignment algorithm, the correct time offset can be identified once a segment of the new stream can be "matched" to the old voice stream using the chosen metric for similarity measure. Many metrics have been developed by the DSP community for measuring the similarity (distance) between two voice streams including cepstral and spectral distance measures [16]. As a proof of concept, in this paper, we use a simple time alignment algorithm based on the cross-correlation of the two voice streams. In essence, a segment of the new stream is matched against segments of the old stream of the same size for finding the time offset that yields the highest cross-correlation. The new stream is then shifted in time accordingly before being "glued" to the old voice stream and sent to the audio output device.

To detail, consider Fig. 5 that shows the old and new voice streams entering the *Audio Mixing* module. Let $r(n)$ and $s(n)$ be the old and new voice streams, respectively, and let $n$ be the time index of audio samples. Assume that the new stream is received at $n = T$, and hence, $s(n) = 0$ for $n < T$. The (normalized) cross-correlation $X(\tau)$ for a time offset of $\tau$ between $r(n)$ and $s(n)$ using segments of $N$ samples can be expressed as

$$X(\tau) = \frac{\sum_{n=T}^{T+N-1} [r(n+\tau) - \bar{r}_N][s(n) - \bar{s}_N]}{\sqrt{\sum_{n=T}^{T+N-1} [r(n+\tau) - \bar{r}_N]^2} \sqrt{\sum_{n=T}^{T+N-1} [s(n) - \bar{s}_N]^2}}, \quad (1)$$

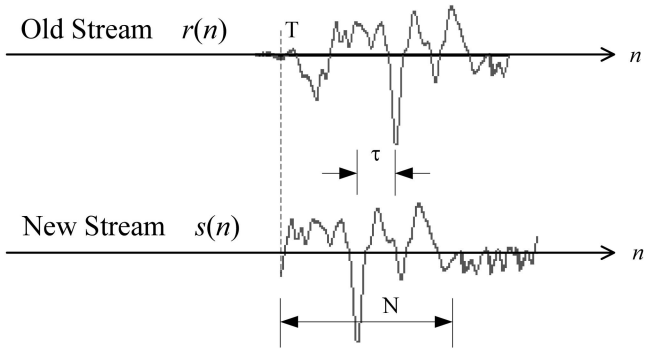


Fig. 5. Cross-correlation for time alignment.

where

$$\bar{r}_N = \frac{1}{N}\sum_{n=T}^{T+N-1} r(n+\tau) \quad \text{and} \quad \bar{s}_N = \frac{1}{N}\sum_{n=T}^{T+N-1} s(n) \tag{2}$$

are the means of the corresponding segments. Let

$$\tau^* = \arg \max_{-L\le\tau\le U} X(\tau) \tag{3}$$

be the time offset (in the search range $-L \le \tau \le U$) that attains the maximum cross-correlation. If $\tau^* > 0$ (i.e., the new voice stream arrives earlier than the old stream), then the audio output $z(n)$ of the *Audio Mixing* module can be expressed as

$$z(n) = \begin{cases} r(n), & \text{if } n < T + \tau^* + \epsilon, \\ A \cdot s(n-\tau^*), & \text{if } n \ge T + \tau^* + \epsilon, \end{cases} \tag{4}$$

where

$$A = \left(\sum_{n=T}^{T+N-1} |r(n+\tau^*)|\right) \Big/ \left(\sum_{n=T}^{T+N-1} |s(n)|\right) \tag{5}$$

is introduced to compensate for gain mismatch between the two voice streams so transition from $r(n)$ to $s(n)$ does not cause perceptible change in magnitude. The parameter $\epsilon$ is introduced to account for the algorithmic delay (e.g., to wait for at least $N$ sample time) and computation time. Note that the computation complexity of the cross-correlation operation depends on the search range $(L+U)$ and granularity (which controls the minimum stepping of candidate time offset $\tau$), as well as the size of the segment $(N)$. Since the time difference between the two voice streams is related to the mismatch in end-to-end delays, information of coarse end-to-end delay estimates of the two networks (e.g., through preset configuration or probing histories) can also help in reducing the search range and, hence, the computation complexity.

While we show in Section 5 that the cross-correlation based time alignment algorithm with proper capping of the search range can achieve desirable performance in the target environment, other sophisticated time alignment algorithms can also be used. For example, in [19], the authors develop a time alignment algorithm that uses envelope cross-correlation for crude estimate of the time difference followed by a fine-scale estimate based on the weighted histogram of the crude delay estimates for individual frames. Matching over a series of segments (instead of only one segment) through dynamic programming techniques [20] can also be used to avoid potential misalignment due to the limited information carried by one segment.

### 4.3 Time Scaling

As we have shown in (4), if the new stream is ahead of the old stream, a simple switching with the help of time alignment and gain control algorithms can achieve seamless handoffs. In the case when the new voice stream lags behind, however, a simple switching will potentially introduce an audio gap between the two voice streams. Such an audio gap can be larger than 300 ms, as shown in Fig. 3. Obviously, one way to hide the audio gap is for the *Audio Mixing* module to wait until the silence period of conversations to switch from the old stream to the new one (as opposed to switching at $T + \tau^* + \epsilon$ in (4)). However, it is not always desirable to wait for the end of the talkspurt before switching. When such an audio gap is exposed to the user, it introduces temporal discontinuity impairments to speech and impacts the perceived quality of speech. It has been shown in [21] that temporal discontinuity impairments degrade the speech quality, and the larger the impairment magnitude (duration of the audio gap) is, the more severe the quality degradation becomes. Moreover, temporal discontinuity impairments introduce semantic and/or syntactic damages to speech, with the damages being more objectionable and annoying to native listeners than to non-native listeners (i.e., when the speech is understood by the listeners) [22].

In this paper, we use an approach based on time-scale modification of speech to address the problem of potential temporal discontinuity during handoffs. Time-scale modification refers to changing the reproduction rate of a signal, and it includes time-scale extension (slow down the playout speed), as well as time-scale compression (speedup the playout speed). Changing the playout speed exploits *human's insensitivity to minor modulations in the speed of a speech signal* [21]. It has been used in a variety of applications including fast listening to voice mail messages and slow playback of the dictation tape. Recently, it has also been used in packet voice networks for addressing the problem of packet losses and network congestion [23], [24].

A key challenge with scaling a speech signal in time is the inherent reverse scaling in the frequency domain that results mathematically from the duality between the frequency domain and the time domain representations of the signal (e.g., pitch being raised due to faster playback speed). A good time scaling algorithm therefore needs to ensure that the perceived timing attribute such as speaking rate is scaled without affecting the *perceived frequency attribute* such as pitch. Many DSP algorithms have been proposed for performing efficient and high-quality time-scale modification of speech [25], [26], [27]. In this paper, we adopt the waveform similarity overlap-add (WSOLA) algorithm [27] for time-scale modification of speech, so the performance degradation due to temporal discontinuity during handoff can be mitigated. The WSOLA algorithm preserves the pitch period and requires *only time domain operations of the speech* without any frequency domain transformation. It has been shown to be computationally efficient, allowing for real-time processing of speech. In the following, we briefly discuss the operations of WSOLA and explain how it can be used for achieving seamless handoffs in the target scenario.

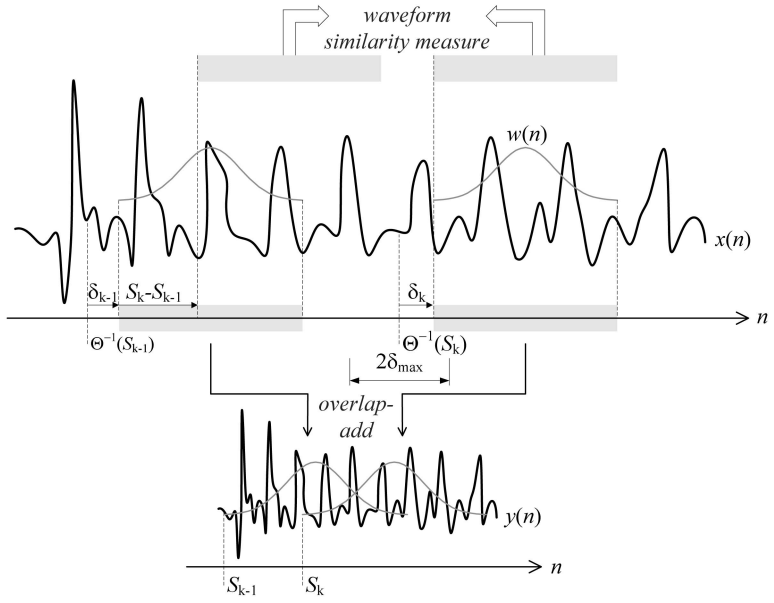


Fig. 6. WSOLA for time-scale modification.

Let $x(n)$ be the original speech signal, and $y(n)$ be the time-scaled version of $x(n)$. Let $\Theta(n)$ be the time warping (scaling) function specifying that the sample occurring at $n$ in $x(n)$ should occur at $\Theta(n)$ in $y(n)$. For example, a linear time scaling operation uses $\Theta(n) = \beta n$ with $\beta < 1$ for compression and $\beta > 1$ for extension of the original signal. The operation of the WSOLA algorithm is based on overlap-add synthesis that consists of cutting out segments of the input signal $x(n)$ around analysis instants $\Theta^{-1}(S_k)$ and repositioning them at corresponding synthesis instants $S_k$ before adding them together to form the output signal $y(n)$. As shown in Fig. 6, if a window function $w(n)$ is used to form analysis segments, then $y(n)$ can be expressed as

$$y(n) = \frac{\sum_k w^2(n - S_k)x(n + \Theta^{-1}(S_k) - S_k + \delta_k)}{\sum_k w^2(n - S_k)} \tag{6}$$

$$= \sum_k w^2(n - S_k)x\big(n + \Theta^{-1}(S_k) - S_k + \delta_k\big), \tag{7}$$

$$\text{if } \sum_k w^2(n - S_k) = 1,$$

where the summation $\sum_k$ is taken to include overlapping segments (refer to Fig. 6) at synthesis instant $n$. If $\delta_k = 0$, then the algorithm simply excises individuals segments from $x(n)$ at $\Theta^{-1}(S_k)$ and shifts them to $S_k$ to form $y(n)$. The problem with this simple approach, however, is that it can introduce pitch period discontinuities and phase jumps at segment joins [28]. The WSOLA algorithm introduces the parameter $\delta_k$, so the synthetic signal $y(n)$ can maintain *maximal local similarity* to the original signal $x(n)$ across segment joins. To find the $k$th synthesis segment in Fig. 6, for example, WSOLA searches in $x(n)$ over a tolerance region $[-\delta_{\max}, \delta_{\max}]$ around $\Theta^{-1}(S_k)$ the segment that has maximal similarity to the right-shifted (by $S_k - S_{k-1}$) segment of the $(k-1)$th synthesis segment. In this way, it is ensured that the new synthesis segment will form a *natural continuation* of the previously chosen segment.

As mentioned in Section 4.2, cross-correlation is a useful metric that can be used as a similarity measure. Another simpler similarity measure that does not involve multiplication is the cross-AMDF (average magnitude difference function). Let $X_A(k, \delta)$ be the cross-AMDF used in WSOLA for deciding the $k$th synthesis segment, then

$$X_A(k,\delta) = \sum_{n=0}^{W-1} \big|x\big(n + \Theta^{-1}(S_{k-1}) + \delta_{k-1} + S_k - S_{k-1}\big) - x\big(n + \Theta^{-1}(S_k) + \delta\big)\big|, \tag{8}$$

where $W$ represents the window length. The position of the $k$th segment will be chosen to have an offset $\delta_k$ as

$$\delta_k = \arg \min_{-\delta_{\max} \le \delta \le \delta_{\max}} X_A(k, \delta). \tag{9}$$

Typically, the synthesis instants are regularly spaced with $S_k = kS$, the window function is a Hann window with 50 percent overlap between successive segments $(W = 2S)$, and a linear scaling function with $\Theta(n) = \beta n$ is used for the WSOLA algorithm. Then, (7) can further be simplified as

$$y(n) = w^2(n - (k-1)S)x\left(n + \frac{(k-1)S(1-\beta)}{\beta} + \delta_{k-1}\right) + w^2(n - kS)x\left(n + \frac{kS(1-\beta)}{\beta} + \delta_k\right), \tag{10}$$

$$\text{for } kS \le n < (k+1)S.$$

If we define $\delta_0 = 0$ and use (9) for deciding $\delta_k$, $k \ge 1$, the time-scaled signal $y(n)$ can be obtained synchronously in a left-to-right fashion.

To apply the WSOLA time scaling algorithm to the target problem, the old voice stream $r(n)$ is extended in time to fill the gap $\tau^*$ (refer to (4)) between the two voice streams. Specifically, let $\beta^\dagger > 1$ be the predetermined scaling factor (at which the change in playout speed is not noticeable by the user). After the new stream arrives at $T$ and the time alignment algorithm finds the time alignment point, if it is

determined that $\tau^* < 0$ (the new stream lags behind), then the time scaling algorithm (WSOLA) starts to expand $r(n)$ synchronously while keeping the new voice stream in the buffer (refer to Fig. 4). WSOLA operates on $r(n)$ until the synthetic stream $y(n)$ overlaps with $s(n)$ in time, after which the *Audio Mixing* module switches the audio output to the new stream. The WSOLA algorithm can be applied again on $s(n)$ after switching to ensure that the transition from $y(n)$ to $s(n)$ forms a natural continuation at the switching point if so desired.

### 4.4 Discussion

We note that to fill the audio gap completely using the time scaling algorithm, the required number of audio samples in $r(n)$ is approximately equal to $|\tau^*|/(\beta^\dagger - 1)$. Since a large scaling factor may result in degradation of perceived speech quality, there is an upper bound on the value of $\beta$ to ensure that the gap is not "closed" at the expense of quality degradation of the old voice stream. If we assume that the delay mismatch between the two voice streams is 300 ms (refer to Fig. 3), and the scaling factor is set to 1.25, then an audio block of 1.2 seconds is needed to fill the audio gap. That is, audio switching cannot be performed until 1.2 seconds after the new stream arrives. While such an additional delay is not significant compared to the call setup time of the GSM audio (refer to Table 1), it might still be desirable to reduce such a delay. A smaller delay, for example, can also reduce the overlapping time of the two voice streams, and hence reducing the extra power consumption for performing "make-before-break" handoff. One possibility is to extend the voice stream before the new stream arrives (recall that the call setup time of the new GSM stream is on the order of 10 seconds). This approach, however, requires the support from the network layer for a raw estimate of the end-to-end delay of the VoWLAN call. In the following, we propose two directions exploiting the linguistic property of speech and human's perception of speech for reducing the delay:

1. It is not necessary that the audio gap is filled up completely. In fact, as shown in [21], the degradation of speech quality due to temporal discontinuity impairments is proportional to the magnitude of the impairment, and the dispersion of the temporal discontinuity is preferable to the clustering of these impairments. Therefore, it is possible that the time scaling algorithm introduces short sparse audio gaps in the synthetic signal to reduce the required length of the input signal. For example, if an audio gap of 5 ms is introduced for every 50 ms of speech, then for a scaling factor of 1.25, only 0.8 second of audio block is needed.
2. While we have used a linear time warping function for time scaling, it is possible to employ nonuniform time scaling techniques to allow for a larger extension ratio without significantly impacting the speech quality. The concept behind nonuniform time scaling of speech is that different speech sounds exhibit different resilience to time scale modification. In [29], the authors classify input speech segments into five classes including pause, plosive-like, vowel-like, consonant-like, and phone transition. Each class is assigned a different scaling factor by taking into consideration the acoustic property of each class. For example, in speech extension, vowels will be stretched more than consonants, and pauses are stretched most. In speech compression, on the other hand, consonants are speeded-up more than vowels, and pauses are speeded-up most. By mimicking the speedup or slowdown strategy of the human speaker, it is possible to have a larger scaling ratio without suffering from the quality degradation that may incur in linear (uniform) scaling.

Further optimization of the algorithm, however, is beyond the scope of this paper. Nonetheless, we have substantiated our argument that *employing DSP techniques in tandem with acoustic and linguistic knowledge of speech* during vertical handoffs is a promising direction of research for achieving seamless voice communications across heterogeneous telephony systems.

## 5 Performance Evaluation

In this section, we present results to show the performance of the proposed solution. We first describe the testbed setup and evaluation metrics and then discuss the selection of related parameters for reducing the computation complexity of the DSP algorithms. Finally, we incorporate the DSP algorithms in the proposed vertical handoff framework and present the evaluation results using testbed experiments.

### 5.1 Testbed Setup

We use a laptop computer with two wireless interfaces as a dual-mode mobile device. The laptop is equipped with a 1.5-GHz CPU, and runs an open source SIP user agent [30] under the Windows XP operating system. The built-in 802.11 b/g wireless access on the laptop is used as the Wi-Fi mode, while an external GSM PCMCIA card is used as the GSM mode. Note that on the laptop in addition to the built-in audio jacks (connected to the internal audio device), the GSM card has another pair of audio jacks for connecting to an external headset (including the earphone and the microphone). To allow the user of the dual-mode laptop to use one only pair of headset (connected to the jacks of the internal audio device) for communicating with the remote SIP phone user, we plug-in a USB audio card on the laptop and connect the audio jacks of the GSM card and the USB audio card, as shown in Fig. 7. The audio I/O of the external USB card (from/to the GSM card) is bridged to the internal audio card using the Windows waveform audio API. In this way, the voice received from the GSM call can pass through the *Audio Mixing* module (refer to Fig. 1), and the voice uttered to the headset can also be received by the *GSM Phone* module to be transmitted to the remote peer. While the setup may seem involved, we note that existing dual-mode handsets with built-in GSM and Wi-Fi modules already have similar hardware design to connect the audio paths from different devices to the common audio I/O. Hence, such external *cabling* will not be necessary on mobile devices with built-in support for dual-mode communication.

As shown in Fig. 8, the SIP infrastructure in our testbed setup consists of a SER SIP proxy server and a Cisco SIP-PSTN gateway. The dual-mode notebook initially has access to campus Wi-Fi networks, which it uses to setup the connection with a remote SIP phone (connected to wired networks). A vertical handoff to the nationwide GSM

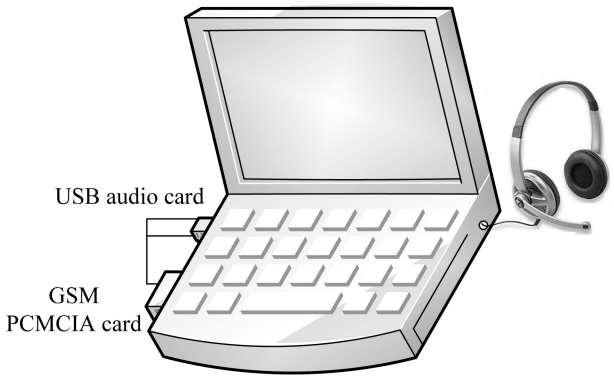


Fig. 7. Dual-mode notebook.

service is initiated when the dual-mode notebook moves outside the coverage of the campus wireless network. The goal of the performance evaluation in this section is to observe the performance when the call is migrated from the Wi-Fi mode to the GSM mode with and without the proposed DSP algorithms.

### 5.2 Evaluation Metrics

We evaluate the performance of the proposed solution in two parts: first, we evaluate the selection of related parameters such that the DSP algorithm achieves the desired performance while incurring low computation complexity; second, we use the chosen parameters for time scaling operations and evaluate the proposed solution for vertical handoffs on the dual-mode device. The first part is evaluated with the help of the perceptual evaluation of speech quality (PESQ) metric for an automated and objective evaluation of parameter selection, and the second part is evaluated using the subjective mean opinion score (MOS) metric for getting true user experience on the overall solution. While MOS is a well-known metric for evaluation of speech quality, PESQ is a relatively new one. In the following, we briefly discuss the PESQ model and how it evaluates the speech quality.

PESQ is an objective method for end-to-end speech quality assessment as recommended by ITU [31]. Unlike its predecessors, PESQ takes into account of filtering, variable delay, coding distortions, and channel errors that may occur in a communication system. As shown in Fig. 9, the PESQ model compares an original signal $x(n)$ with a degraded signal $y(n)$ using *perceptual and cognitive models*. First, the input signal is transformed into internal representation resembling the psychophysical representation of audio signals in the human auditory system (e.g., frequency warped to pitch scale and intensity warped to loudness scale). Several stages of processing such as time alignment, level alignment, time-frequency mapping, frequency warping, and compressive loudness scaling are included in the perceptual model. The time alignment algorithm, similar in concept to that presented in Section 4.2, computes a series of delays between original and degraded signals, one for each time interval for which the delay is significantly different from the previous time interval. Thus, it can handle the case with variable delays (piecewise constant delay) during silences and during active speech parts [19]. Afterwards, the PESQ model compensates for minor impairments that may have little perceptual significance such as local gain variations and linear filtering distortions. More severe effects are only partially compensated so that a residual effect remains and contributes to the overall perceptual disturbance. The PESQ model calculates and weighs disturbances caused by different components separately before they are combined by the cognitive model to give an objective listening quality score. The final PESQ score is obtained by mapping the objective score to the subjective score (using a large set of subjective experiments) through regression. The range of the PESQ score is from $-0.5$ (worst) to 4.5 (best). Benchmark results including tests for transmission channel errors, effect of varying delay in listening, and time warping of audio signal have shown *a high correlation (0.935) between PESQ and subjective scores* for both known and unknown data [31].

We note that compared to the $R$ *Factor* (Transmission Rating Factor) derived from the ITU E-Model [32], PESQ does not rely on the calculation of various transmission

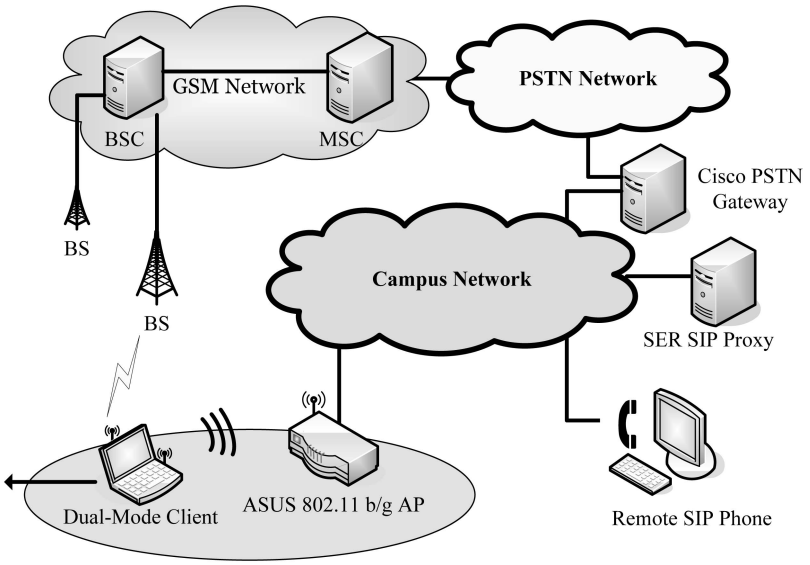


Fig. 8. Testbed setup.

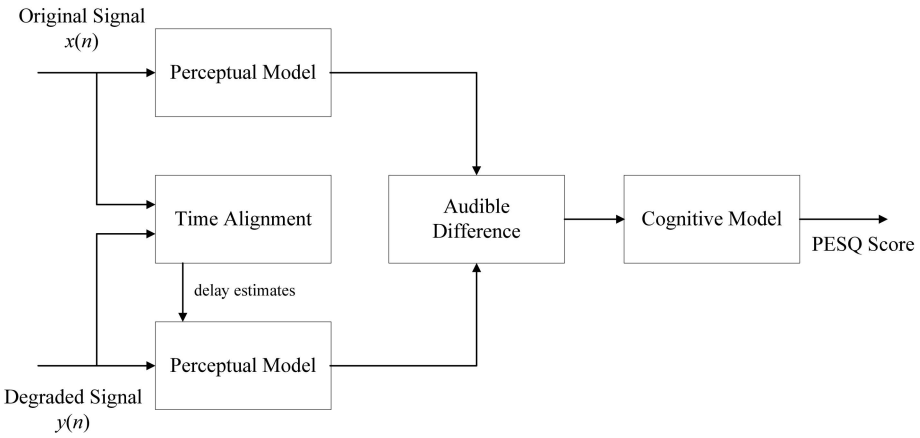


Fig. 9. PESQ model [31].

parameters such as end-to-end delay, packet losses, and codec types and their transformations to impairment factors for predicting the quality of the call [33]. For handoffs across heterogeneous telephony systems, since the GSM voice is not packetized as the VoWLAN voice, PESQ manifests itself as a better candidate than the *R Factor* for evaluation of speech quality.

## 5.3 Evaluation Results

In this section, we investigate the computation complexity (execution time) of the WSOLA algorithm and discuss how such complexity can be effectively reduced through appropriate selection of parameters. We then present the performance of the proposed solution based on the chosen parameters during testbed experiments.

### 5.3.1 Parameters Selection

While the ability of WSOLA to produce high-quality time-scale modified speech has been shown in related work [27], to be used feasibly as part of the handoff operation, its computation complexity is an important factor. We proceed by conducting experiments as follows: first, we extend a segment of speech $x(n)$ by a ratio of $\beta$ to get $x'(n)$, and then compress the extended speech $x'(n)$ by a ratio of $1/\beta$ to get $y(n)$. The output speech $y(n)$ thus has the same duration (and timing attribute) as the original speech $x(n)$, but it can be considered as a *degraded speech* due to the operations of WSOLA using the control parameter. The impact of varying the control parameter on the performance of the WSOLA algorithm thus can be profiled by the change in the PESQ score of $y(n)$ against $x(n)$. Related work has shown that such a methodology can perform well in evaluating different time-scale modification algorithms [34].

To reduce the computation complexity of the WSOLA algorithm, we fist vary the length of the Hann window $W$ and investigate its impact on speech quality. The computation complexity is measured as the execution time of the algorithm normalized to the duration of speech, and hence, a value of 1 or less is required for the algorithm to run in real time. We use $\beta = 1.25$ and vary the window length from 10 ms to 200 ms and plot the results in Fig. 10a. It can be shown from the figure that as the window length decreases from 200 ms to 10 ms, the execution time also decreases from 1.55 to 0.1 real time. A small window length is thus needed to reduce the computation complexity. Note that in the WSOLA algorithm, the synthesis instants are spaced by $W/2$, and hence, the larger the window length, the coarser the spacing of the synthesis instants. It can be observed from the figure that reducing the window length has the effect of improving the speech quality until $W = 40$ ms. If the window length decreases below 40 ms, then the quality starts to decrease sharply. This is because if the window length is too small, the analysis segment does not contain sufficient information to reliably perform the similarity matching operation in WSOLA. Tests with different speech contents and speakers (including male and female speakers) show similar results as indicated in Fig. 10b. It can be observed that there exists a lower bound of the window length above which the computation complexity of WSOLA can be significantly reduced without degrading the speech quality. As Fig. 10b shows, the average peak value stays at around 40 ms for the tests conducted.

To further reduce the computation overheads, it is therefore not desirable to keep decreasing the window length. Alternatively, note that at each synthesis instant WSOLA searches within a range (tolerance region) $[-\delta_{\max}, \delta_{\max}]$ for finding the "most similar" analysis segment. For a given window length, the computation complexity thus can be reduced through control of the search range (i.e., $\delta_{\max}$). We start with a window length of 40 ms and vary the search range from 1/3 to 1/30 of the frame length (i.e., the "delta divisor" varies from 3 to 30). Fig. 11a shows the impact of varying the search range, with results averaged over three different speakers, as in Fig. 10b. It can be observed from the figure that reducing the search range does decrease the computation complexity: the normalized execution time decreases from 0.66 to 0.07 real time when the divisor varies from 3 to 30. The computation reduction is, however, at the expense of degradation in the speech quality (from 3.7 to 3.1). The degradation of speech quality obviously is due to the possibility of finding a "less similar" waveform at each synthesis instant when the search range decreases. Note that in the WSOLA algorithm, we can obtain the "waveform similarity" measure (e.g., the inverse of cross-AMDF) at each synthesis instant and plot the average value of the similarity measure for the entire speech for each delta divisor value. As shown in Fig. 11a, waveform similarity

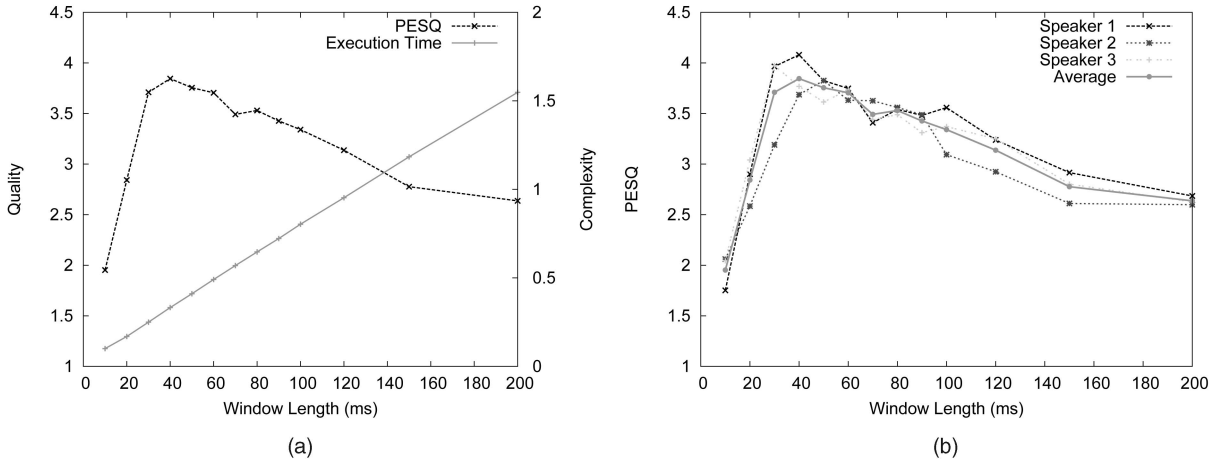


Fig. 10. Impact of window length. (a) Quality versus complexity. (b) Different speakers.

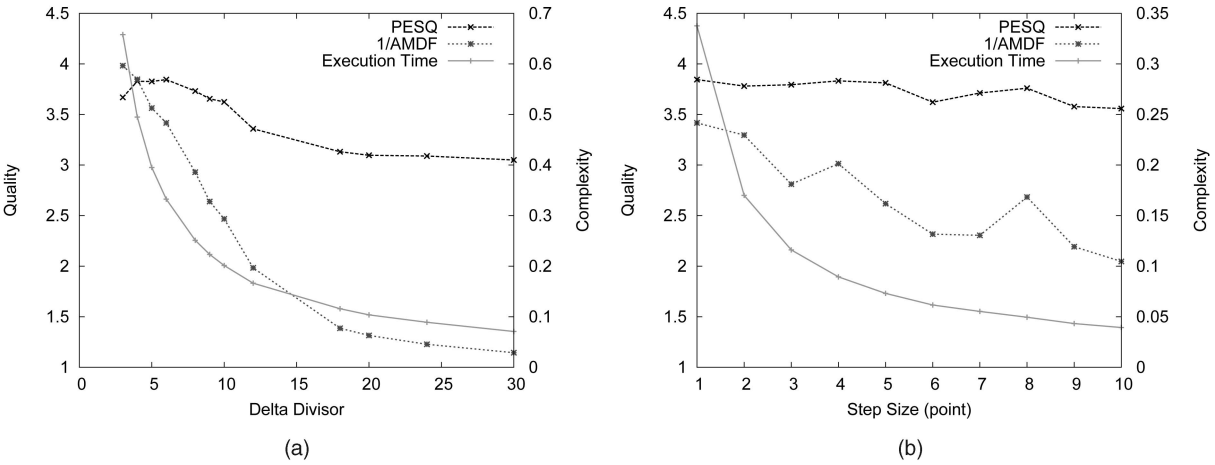


Fig. 11. Impact of search range and granularity. (a) Search range (delta divisor). (b) Search granularity (step size).

(with proper normalization to the PESQ scale for clarity) drops as the delta divisor increases. This observation substantiates the argument that decreasing the search range does impact waveform similarity in a nonincreasing fashion. Moreover, it can be observed that in the WSOLA algorithm, *waveform similarity has a good correlation to overall speech quality*.

In addition to controlling the search range, another way to reduce the computation complexity of the WSOLA algorithm is to control the *search granularity*. That is, for a given search range, the search for best match does not need to be performed on a sample-by-sample basis but on a coarser granularity. To investigate the impact of increasing the search granularity, we use a 40 ms window length and fix the delta divisor at 6. Fig. 11b shows the impact of varying the search granularity (step size). The control of search granularity has a similar effect to the control of search range. When the step size increases from 1 to 10, for example, the execution time decreases from 0.34 to 0.04 real time, while the speech quality decreases from 3.8 to 3.5. Note that a 1/10 times decrease in the search range has a similar impact on computation complexity to a 10 times increase in the search granularity. The degradation in speech quality by controlling the search granularity, however, is slightly less than that by controlling the search range.

To compare the effectiveness of decreasing the window length, decreasing the search range, and increasing the search granularity in reducing the computation complexity, we plot the quality-complexity trade-offs in Fig. 12a. We use the data for obtaining the PESQ curves in Figs. 10b, 11a, and 11b, but plot the curves against the normalized execution time (as opposed to the control parameters). The execution time is normalized to the case when the window length is 40 ms, delta divisor is 6, and step size is 1. It can be observed from the figure that, compared to the baseline case increasing the step size is most effective in reducing the computation complexity while incurring minimal degradation of speech quality. Fig. 12b compares the speech quality and execution time for several different parameter sets. It is obvious that to reduce the computation complexity, related WSOLA parameters need to be properly chosen to avoid significant impact on speech quality. In the following, we use a window length of 40 ms, a delta divisor of 6, and a step size of 5 for employing the WSOLA algorithm during handoff. We note that the resultant configuration, as shown in Fig. 12b, requires only 0.07 times of speech duration to perform time-scale modification of speech. While the experiments have been performed on a 1.5-GHz notebook, existing dual-mode handsets have been equipped with CPUs of several hundred MHz. Therefore, it is feasible to run the proposed solution for handoff in

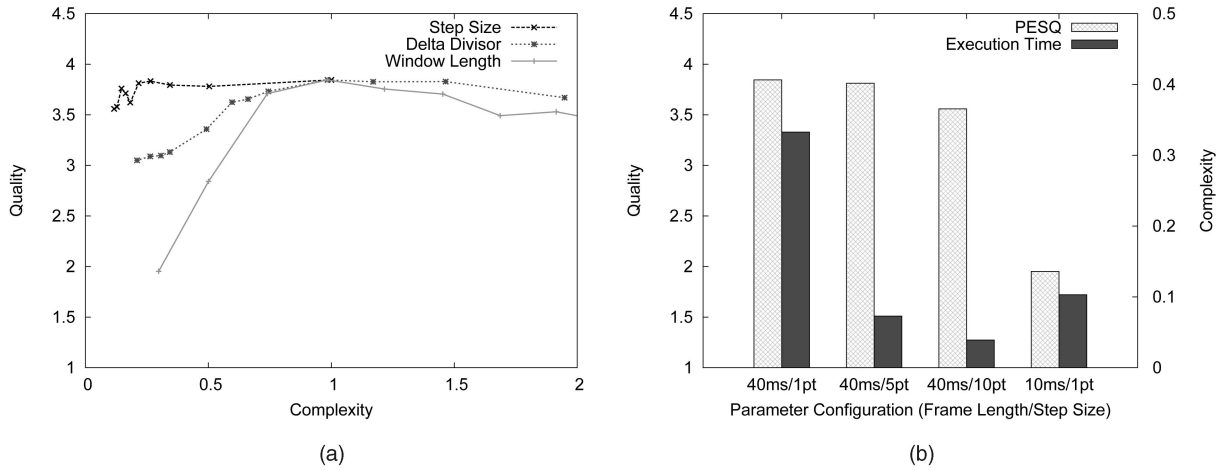


Fig. 12. WSOLA parameter optimization. (a) Effectiveness of quality-complexity trade-offs. (b) Overall comparison.

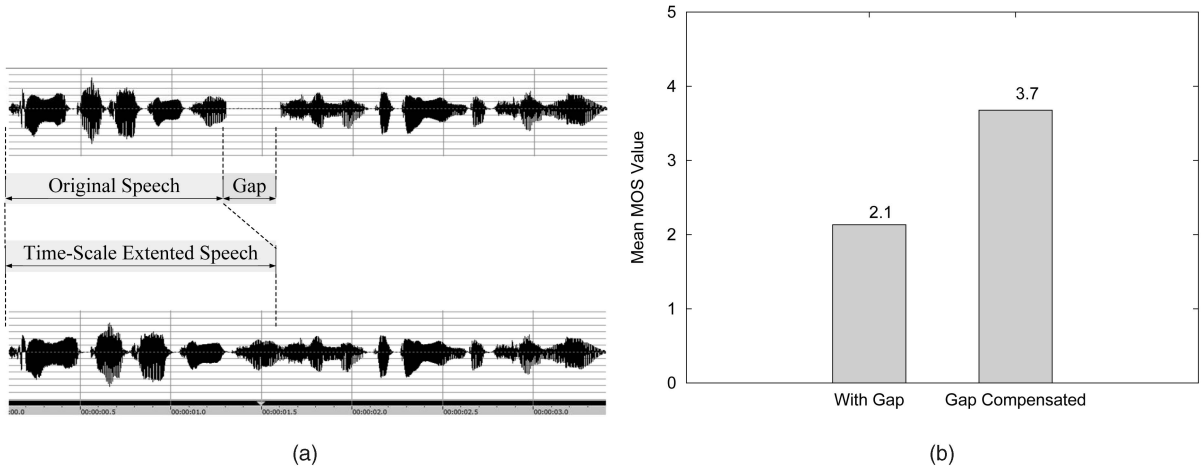


Fig. 13. Seamless call transfer. (a) Waveform inspection. (b) Mean opinion score.

tandem with the signal processing algorithms on modern dual-mode handsets.

### 5.3.2 Seamless Handoffs

We have investigated in Section 5.3.1 the impact of various parameters in the computation complexity of the algorithm. In this section, we show how the algorithm performs with the chosen parameters in achieving seamless vertical handoffs on dual-mode mobile devices.

As mentioned in Section 5.1, we consider a scenario when the dual-mode notebook was initially in a VoWLAN call with the remote SIP phone, and then the call is migrated from the Wi-Fi mode to the GSM mode. We evaluate the speech quality on the dual-mode notebook with and without the proposed DSP algorithms. Note that in the target environment, the delay mismatch between the VoWLAN call (through the campus Wi-Fi network), and the GSM call (through the national GSM network) is about 300 ms. If no DSP algorithm is used to mitigate the audio gap, then the gap is exposed to the user during the handoff as shown in Fig. 13a. If the proposed algorithm is used to extend the VoWLAN voice stream, on the other hand, the gap between the two audio streams can be reduced effectively, as shown in Fig. 13a.

To evaluate the real impact of the proposed solution on user experience during handoffs, we conduct an experiment with 30 listeners. Each listener is presented with the aforementioned two voice streams, one with an audio gap of 300 ms, and the other with the gap compensated by the proposed solution. The listener is asked to compare the quality of the two voice streams and then give a score from 1 to 5 for the two voice streams. Fig. 13b thus shows the results of the tests. It can be observed that while the proposed solution in essence introduces artifacts (time-scale modification) to the original speech, the impact on user experience is not significant—not noticeable even to some listeners. Users would prefer a voice communication that is continuous (possibly prosodic with natural speed variation), rather than one with unpleasing temporal discontinuity. Therefore, we conclude that the proposed solution does provide a more pleasing experience during vertical handoffs.

## 6 Related Work and Conclusions

In this section, we discuss and compare related work for handoff support on dual-mode mobile devices, and then conclude the paper.

### 6.1 Related Work

While network heterogeneity has attracted a lot of attention recently, the problem considered in this paper is quite

different from that in related work due to the nature of the telephony systems involved. Related work on seamless handoffs between heterogeneous wireless networks has focused on data access using IP as the underlying platform [4], [5], [6], [7], [10]. This paper, on the other hand, considers *voice communications across circuit-switched (GSM) and packet-switched (WLAN) networks*. Note that GSM uses a very different protocol stack from the WLAN in terms of both signaling and data transport. For example, GSM circuit-switched voice, after proper digitalization, coding, and interleaving, is fitted into the TDMA frames for transmissions, while WLAN packet-switched voice undergoes an entirely different protocol stack including codec selection, RTP packetization, UDP multiplexing, IP switching, and 802.11 MAC framing. Hence, conventional approaches that rely on all-IP solutions (e.g., Mobile IP or TCP based approaches) cannot be used in this context. While it is possible to support voice communications through GPRS packet data access (VoIP over GPRS), it is not practical due to the apparent lack of competitiveness in cost and quality compared to the GSM circuit-switched voice service.

As we mentioned in Section 1, several dual-mode handsets are released to support handoffs between the two modes without user intervention. However, they have been designed for use with customized support from the GSM or WLAN infrastructure. For example, Motorola's CN620 dual-mode handsets rely on the MAP architecture developed jointly by Motorola, Avaya, and Proxim to provide seamless handoffs in target environments [35]. Azaire Networks Inc. develops an "IP Converged Network Platform (IP-CNP)" using a 3GPP-WLAN convergence gateway and teams up with several telecom service providers for deployment. VeriSign's "Wireless IP Connect Service" and Calypso Wireless' ASNAP solution are yet another two examples.[4] Apparently, these proprietary solutions are tailored to specific dual-mode handsets and service providers. Hence, such dual-mode handsets cannot be used in systems operated by different service and equipment providers. The interoperability of different incompatible systems will eventually become an issue hurdling the popularity of such dual-mode handsets.

Recently, an open standard called Unlicensed Mobile Access (UMA) has been developed by the UMA Consortium. It is later adopted by the 3rd Generation Partnership Project (3GPP) as the specification for *Generic Access (GA) to the A/Gb interfaces* in the public land mobile network (PLMN such as GSM core network) [8]. The goal is to extend GSM mobile service over an IP network (called Generic Access Network or GAN), so mobile stations can obtain services from the GSM core network through IP access rather than through the traditional GERAN (GSM/EDGE Radio Access Network) radio interface. In the GAN architecture, a new network element called GAN controller (GANC) is introduced. To bridge circuit-switched service in the PLMN and packet-switched data in the GAN, GANC performs functionalities, including reframing from RTP packets to circuit-switched frames, transcoding voice to/from the MS to PCM voice from/to the MSC, and establishment, administration and release of control/user plane bearers between the MS and the core network. In addition to changes in the infrastructure, the MS also needs to be upgraded to perform resource control functionalities including discovery, registration, and keep alive with the GANC, as well as circuit-switched functionalities, including setup of bearer for CS traffic between the MS and GANC, and handoff support between the GERAN and GAN. Several cellular telephony service providers and handset manufacturers have announced their plans to support the GAN architecture.

The advantage of the GAN architecture is that it integrates heterogeneous wireless data networks into the PLMN for providing a unified access to the mobile device, and it is possible to use legacy devices such as POTS phones for communications with the dual-mode device. However, there is still motivation for dual-mode users to consider solutions orthogonal to GAN. For one, since all traffic from the WLAN traverses through the GSM system, the dual-mode user will be charged for using service provided by the GSM core network (note that the packet-switched voice from MS is terminated at the GANC) when *placing VoWLAN calls with the purpose to save the cost incurred in using the GSM phone service*. Also, WLAN traffic has to flow through the GSM core network, incurring additional latency and introducing bottleneck to voice communications between the dual-mode user and the remote peer. The problem will be more serious if GANC's are deployed without sufficient density. Finally, with the emergence of new wireless data technology such as WiMax (IEEE 802.16), it is possible that there will be GSM-WiMax or 3G-WiMax handsets in the future. It remains to be seen whether infrastructure-based approaches such as GAN can be easily adapted to support new wireless systems. Still, as we mentioned in Section 1, new functionalities added to dual-mode handsets, as proposed in this paper, do not preclude the adoption of infrastructure-based solutions such as GAN in the future.

### 6.2 Conclusions

In this paper, we investigate the problem of supporting seamless voice communications across heterogeneous telephony systems. Instead of relying on infrastructure-based interworking solutions, however, we consider a solution that can be deployed solely on the end devices. Specifically, we focus on VoIP service based on the SIP, and we explain why existing solutions cannot be used. We identify the key challenges of such an end-to-end solution as the requirement to address latency mismatches between circuit-switched and packet-switched voice calls. Toward this goal, we employ digital speech processing techniques to process the voice streams of the GSM and VoWLAN calls for achieving seamless call transfers. We conduct testbed experiments using a GSM-Wi-Fi dual-mode notebook and investigate the voice quality when the call is migrated from Wi-Fi to GSM modes. Evaluation results show that such a cross-layer optimization between networking and DSP techniques can effectively address the limitations of existing solutions and achieve the desired performance. Our ongoing work includes porting of the proposed vertical handoff framework and the DSP algorithms to off-the-shelf dual-mode handsets for further investigation into the performance benefits.

4. Detailed information on individual products and solutions from Azaire Networks (http://www.azairenet.com), VeriSign (http://www.verisign.com/), and Calypso Wireless (http://www.calypsowireless.com) is available online.

## ACKNOWLEDGMENTS

The authors would like to thank the editors and anonymous reviewers for their valuable suggestions that helped improve the quality of this paper. This work was supported in part by funds from the Excellent Research Projects of the National Taiwan University under Grant 97R0062-06 and the ROC Ministry of Economy Affairs under the Wireless Broadband Communications Technology and Application Project of the Institute for Information Industry.

## REFERENCES

[1] Apple Inc., *iPhone,* http://www.apple.com/iphone/, 2007.
[2] HTC Corp., *CHT9100,* http://www.htc.com, 2006.
[3] HTC Corp., *Dopod 818 Pro,* http://www.htc.com, 2006.
[4] M. Stemm and R. Katz, "Vertical Handoffs in Wireless Overlay Networks," *ACM/Kluwer Mobile Networks and Applications,* vol. 3, no. 4, pp. 335-350, 1998.
[5] H.-Y. Hsieh, K.-H. Kim, Y. Zhu, and R. Sivakumar, "A Receiver-Centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces," *Proc. ACM MobiCom '03,* pp. 1-15, Sept. 2003.
[6] R. Inayat, R. Aibara, and K. Nishimura, "A Seamless Handoff for Dual-Interfaced Mobile Devices in Hybrid Wireless Access Networks," *Proc. IEEE Int'l Conf. Advanced Information Networking and Applications (AINA '04),* pp. 373-378, Mar. 2004.
[7] H.-H. Choi, O. Song, and D.-H. Cho, "A Seamless Handoff Scheme for UMTS-WLAN Interworking," *Proc. IEEE Global Telecomm. Conf. (Globecom '04),* pp. 1559-1564, Nov. 2004.
[8] *Generic Access to the A/Gb Interface; Stage 2,* Third Generation Partnership Project, 3GPP TS 43.318 V6.7.0, July 2006.
[9] M. Buddhikot, G. Chandranmenon, S. Han, Y.-W. Lee, S. Miller, and L. Salgarelli, "Design and Implementation of a WLAN/CDMA2000 Interworking Architecture," *IEEE Comm. Magazine,* vol. 41, no. 11, pp. 90-100, Nov. 2003.
[10] A. Salkintzis, G. Dimitriadis, D. Skyrianoglou, N. Passas, and N. Pavlidou, "Seamless Continuity of Real-Time Video across UMTS and WLAN Networks: Challenges and Performance Evaluation," *IEEE Wireless Comm. Magazine,* vol. 12, no. 3, pp. 8-18, June 2005.
[11] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, *SIP: Session Initiation Protocol,* IETF RFC 3261, June 2002.
[12] A. Vemuri and J. Peterson, *Session Initiation Protocol for Telephones (SIP-T): Context and Architectures,* IETF RFC 3372, Sept. 2002.
[13] H. Schulzrinne and E. Wedlund, "Application-Layer Mobility Using SIP," *ACM Mobile Computing and Comm. Rev.,* vol. 4, no. 3, pp. 47-57, July 2000.
[14] R. Sparks, *The Session Initiation Protocol (SIP) Refer Method,* IETF RFC 3515, Apr. 2003.
[15] C. Elliott, "Stream Synchronization for Voice over IP Conf. Bridges," master of engineering thesis, McGill Univ., Nov. 2004.
[16] S. Quackenbush, T. Barnwell III, and M. Clements, *Objective Measures of Speech Quality.* Prentice Hall, 1988.
[17] R. Steinmetz, "Human Perception of Jitter and Media Synchronization," *IEEE J. Selected Areas in Comm.,* vol. 14, no. 1, pp. 61-72, Jan. 1996.
[18] S. Weinstein, "Echo Cancellation in the Telephone Network," *IEEE Comm. Magazine,* vol. 15, no. 1, pp. 8-15, Jan. 1977.
[19] A. Rix, M. Hollier, A. Hekstra, and J. Beerends, "Perceptual Evaluation of Speech Quality (PESQ): The New ITU Standard for End-to-End Speech Quality Assessment; Part I—Time-Delay Compensation," *J. Audio Eng. Soc.,* vol. 50, no. 10, pp. 755-764, Oct. 2002.
[20] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993.
[21] S. Voran, "Perception of Temporal Discontinuity Impairments in Coded Speech—A Proposal for Objective Estimators and Some Subjective Test Results," *Proc. Int'l Conf. Measurement of Speech and Audio Quality in Networks (MESAQIN '03),* May 2003.
[22] D.-S. Kim, "ANIQUE: An Auditory Model for Single-Ended Speech Quality Estimation," *IEEE Trans. Speech and Audio Processing,* vol. 13, no. 5, pp. 821-831, Sept. 2005.
[23] Y. Liang, N. Farber, and B. Girod, "Adaptive Playout Scheduling and Loss Concealment for Voice Communication over IP Networks," *IEEE Trans. Multimedia,* vol. 5, no. 4, pp. 532-543, Dec. 2003.
[24] H. Ilk and S. Guler, "Adaptive Time Scale Modification of Speech for Graceful Degrading Voice Quality in Congested Networks for VoIP Applications," *Signal Processing J.,* vol. 86, no. 1, pp. 127-139, Jan. 2006.
[25] R. Roucos and A. Wilgus, "High Quality Time Scale Modification for Speech," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '85),* Apr. 1985.
[26] D. Bigorgne et al., "Multilingual PSOLA Text-to-Speech System," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '93),* Apr. 1993.
[27] W. Verhelst and M. Roelands, "An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '93),* Apr. 1993.
[28] W. Verhelst, "Overlap-Add Methods for Time-Scaling of Speech," *Elsevier Speech Comm. J.,* vol. 30, no. 4, pp. 207-221, Apr. 2000.
[29] M. Demol, W. Verhelst, K. Struyve, and P. Verhoeve, "Efficient Non-Uniform Time-Scaling of Speech with WSOLA," *Proc. Int'l Conf. Speech and Computer (SPECOM '05),* Oct. 2005.
[30] SIPfoundry, *sipXezPhone—A New sipXtapi Based SIP User Agent,* http://www.sipfoundry.org/sipXezPhone/, 2008.
[31] ITU-T, *Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs,* ITU Recommendation P.862, Feb. 2001.
[32] ITU-T, *The E-Model: A Computational Model for Use in Transmission Planning,* ITU Recommendation G.107, Dec. 1998.
[33] L. Atzori and M. Lobin, "Speech Playout Buffering Based on a Simplified Version of the ITU-T E-Model," *IEEE Signal Processing Letters,* vol. 11, no. 3, pp. 382-385, Mar. 2004.
[34] F. Liu, J. Lee, and C.-C. Kuo, "Objective Quality Measurement for Audio Time-Scale Modification," *Proc. SPIE Internet Multimedia Management Systems (IMS '03),* pp. 208-216, Nov. 2003.
[35] Motorola Inc., *Motorola Seamless Mobility,* http://www.motorola.com, 2008.

**Hung-Yun Hsieh** received the BS and MS degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, and the PhD degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta. He joined the Department of Electrical Engineering and the Graduate Institute of Communication Engineering at the National Taiwan University as an assistant professor in August 2004. His research interests include wireless communications, digital speech processing, and mobile computing.

**Chung-Wei Li** received the BS degree in electrical engineering from the National Central University, Jhongli, Taiwan, in 2004, and the MS degree in communication engineering from the National Taiwan University, Taipei, Taiwan, in 2006. His research interests include voice-over-IP technology and wireless communication.

**Hsiao-Pu Lin** received the BS degree in electrical engineering from the National Cheng Kung University, Tainan, Taiwan, in 2006, and the MS degree in communication engineering from the National Taiwan University, Taipei, Taiwan in 2008. His research interests include vertical handoff and audio-video synchronization for dual-mode mobile devices.