A Novel Mobility-Aware Playout Algorithm for VoIP Services

Asimakis Lykourgiotis, [1]✉

Email asly@ece.upatras.gr

Asimakis Lykourgiotis received his Ph.D. in mobile and multimedia networking from the Electrical and Computer Engineering Department of the University of Patras, Hellas, in 2016. He is a postdoctoral research fellow at the School of Science and Technology at the Hellenic Open University and at the Wireless Telecommunications Laboratory of the Department of Electrical and Computer Engineering at the University of Patras. His main research interests include wireless local area networks, cellular networks, emergency services, mobility management, and multimedia transmission over wireless networks. He has been involved in EU projects.

Stavros Kotsopoulos, [1]

Email kostop@ece.upatras.gr

Stavros Kotsopoulos received his B.Sc. in physics in 1975 from Aristotle University of Thessaloniki, Greece, and in 1984 got his diploma in electrical and computer engineering from the University of Patras. He received his M.Phil. in 1978 and Ph.D. in 1985 from the

University of Bradford, United Kingdom. Currently, he is a professor with the Department of Electrical and Computer Engineering of the University of Patras. Since 2004, he has been director of the Wireless Telecommunications Laboratory. His research interests include cellular mobile communications, wireless networks, interference, satellite communications, telematics applications, communication services, and antennae design. Moreover he is the (co)author of the book Mobile Telephony. He has more than 400 publications in journals and conferences, and has been the leader of several international and many national research projects.

Tasos Dagiuklas, [2]

Email tdagiuklas@lsbu.ac.uk

Tasos Dagiuklas is a leading researcher and expert in the fields of Internet and multimedia technologies for smart cities, ambient assisted living, healthcare and smart agriculture. He is the leader of the newly established SuITE research group at the London South Bank University where he also acts as the Head of Division in Computer Science. Tasos Dagiuklas received the Engineering Degree from the University of Patras-Greece in 1989, the M.Sc. from the University of Manchester-UK in 1991 and the Ph.D. from the University of Essex-UK in 1995, all in Electrical Engineering. He has been a principle investigator, co-investigator, project and technical manager, coordinator and focal person of more than 20 internationally R&D and Capacity training projects with total funding of approximately €5.0 m from different international organizations. His research interests include Smart Internet Technologies, Media Optimization across heterogeneous networks, QoE, Virtual Reality, Augmented Reality and cloud infrastructures and services.

[1] Wireless Telecommunications Laboratory, Department of Electrical and Computer Engineering, University of Patras, Patras, Greece

[2] Division of Computer Science and Informatics, London South Bank University, 103 Borough Road, London, UK

Abstract

The latest explosive growth in mobile networks has resulted in an increasing interest in optimisation techniques for mobile services. Resent advances in mobile wireless networks incorporate link-layer intelligence in order to enhance the performance of network and application layers. The Media Independent Handover (MIH) standard provides a framework that can make such link-layer intelligence available to upper layers. In this paper, a novel MIH-enabled playout algorithm for Voice over Internet Protocol applications is presented that aims to compensate for the degradation of voice quality caused by handovers. To that end, link-layer triggers in conjunction with speech time-scale modification techniques are exploited to mitigate the increase in delay and jitter induced by the handover process. Results of subjective listening tests show typical gains of 0.3 on a 5-point scale of the Mean Opinion Score with respect to existing playout scheduling schemes.

AQ1

Keywords

VoIP
Handover management
Playout algorithm
Fast handovers
WSOLA
MIH

1. Introduction

The pervasiveness of Internet technologies to everyday life has greatly increased during the recent years. Additionally, the widespread deployment of mobile devices resulted to rapid expansion of Internet applications over wireless networks. A major challenge for the Future Internet is to support ubiquitous mobile connectivity with guaranteed quality of experience. As a result, mobility management becomes a crucial aspect of the migration towards all-IP networks. Especially real-time applications, such as VoIP and video streaming services, impose strict limitations upon the performance of mobility management. Considerable effort has been made to minimise the handover delay in order to meet these limitations. Various mobility management protocols [19, 22] and extensions [7, 12, 25] have been proposed for that purpose, operating at different layers of the protocol stack. In all proposed solutions, the handover process can lead to disruption of communication [1, 14].

Handover is the process by which a Mobile Node (MN) changes its point of attachment as it moves across different wireless networks. Handling mobility in IP-based networks regards the maintenance of active sessions, while the end-user moves across different subnets. In the context of ubiquitous connectivity, it is important to define the framework that will allow mobility protocols to execute handovers uniformly among heterogeneous link-layer interfaces. MIH [8] protocol is such a framework that can handle the upcoming challenges of seamless mobility and session continuity. MIH is a standard that can support link-layer intelligence to upper layers. The introduction of MIH is based on the results of intense research on cross-layer design and handover coordination in heterogeneous networks [3, 17]. This mainly involves optimisations in network selection and handover process. Additionally, many mobile applications have complex functionalities (e.g., playout scheduling) that need such intelligence to meet their aims [2, 4].

An other challenge in supporting interactive voice over the Internet is to compensate for the varying network conditions, such as delay and jitter. This involves a playout buffer mechanism that efficiently schedules the playout time of voice packets. Playout schemes can be grouped in two categories: fixed and adaptive [21]. Fixed playout schemes use a fixed playout deadline for all voice packets throughout the duration of a voice call. Contrarily, adaptive playout scheduling adjusts the playout delay to the varying network conditions. Commonly, in adaptive playout scheduling schemes, playout time is adaptively adjusted during silence periods. This approach reacts slower to a change in the network conditions. Alternatively, time-scale modification techniques have been proposed to adjust the playout schedule of each individual packet [11, 13]. These schemes are more reactive and robust to delay variations as they adjust the packet duration to the current network conditions in a highly dynamic manner. Further optimisation is possible to playout scheduling during handovers by reaping the benefits of link-layer intelligence. Thus, it is crucial to investigate the interaction between MIH and VoIP.

Optimisations of VoIP application with regards to mobility have been studied in the past. In [2], the benefits of coupling mobility management with VoIP services were investigated, mainly focusing on fast adaptation on the conditions of the new network after the handover. This approach involves content adaptation in order to fulfil the bit rate constrains of the new network by session or audio codec adaptation. It also proposes that the playout scheduling algorithm should exploit link-layer intelligence to better estimate new network conditions. In [4], an opportunistic vertical handover scheme based on the on–off characteristics of VoIP traffic is proposed. This scheme aligns the handover process with the mutual silence periods (i.e., both parties engaged in the conversation are silent) whenever possible.

As discussed above, previous works in the field of mobility management with regards to VoIP applications have proposed cross layer architectures. However, they did not consider the playout mechanism in their solutions. Most of the scientific effort in the field of mobility management, considers only the network metrics. This oversight results to packet losses at the application layer which, as we will demonstrate, can be avoided. In particular, in our previous work on fast handovers [15], we have studied cross layer optimizations between the link layer and the network layer, in order to minimize network packet loss rate. However, it became apparent that this reduction in the network layer is not always reflected in the application layer. As a result, the resources that have been reserved by the fast handover protocols to buffer packets during the handover and forward them after its completion are wasted. Thus, in the current research work, we propose a cross layer architecture that includes the full protocol stack from the link layer to the application layer. For that purpose, we make available link layer information to a novel playout algorithm that, in contrast with previous works, does optimization in a per packet basis at the application layer and thus achieving greater overall performance, as it is shown by the results. Specifically, our work investigates possible optimisations when the handover inevitably occurs during a talkspurt. For this purpose, we propose a novel playout scheduling scheme that exploits link-layer events and time-scale modifications during handovers. By scaling individual voice packets during handovers, less distortion is introduced and less packets are being dropped. Moreover, link-layer intelligence provides adequate time to conceal the disruption caused by the handover process. To the best of our knowledge, the proposed algorithm is the first that focuses on time-scale modifications for playout scheduling in accordance with the handover process to eliminate the introduced distortion.

This paper is organised as follows. Section 2 outlines the basic concept of mobility management and link-layer intelligence. Section 3 briefly describes playout delay algorithms and some basic packet loss concealment techniques. In Sect. 4 the proposed adaptive playout scheduling is described in depth. Section 5 describes the methodology used in the subjective quality tests and presents the obtained results. Finally Sect. 6 concludes the paper.

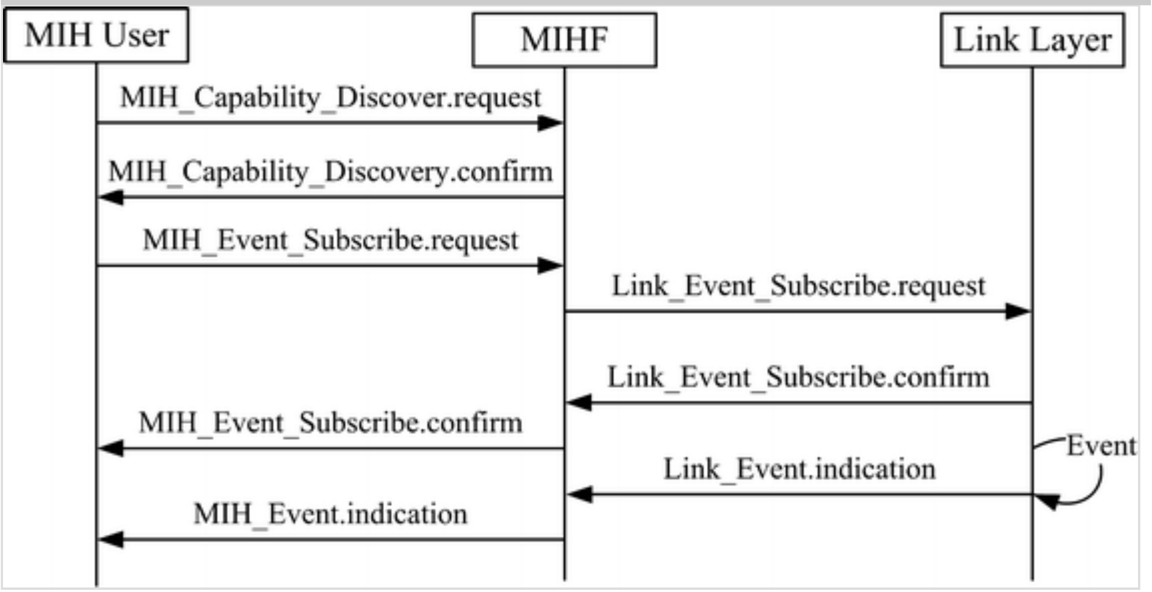
2. Mobility Management

Mobility management is the process of keeping track of the location of the MN between successive calls (location management), and supporting service continuity when the MN changes its point of attachment to the network (handover management). Regarding the involved technologies, a handover can be categorised either as horizontal or vertical [18]. A horizontal handover denotes a change in the point of attachment to the network within the same access technology, while a vertical handover a switch between different access technologies. In terms of connectivity, a handover can be of two types namely hard or soft. A hard handover is a *break before make* process where the MN first disconnects from the old point of attachment and then connects to the new one. During this period the MN has no connectivity to the network and packets destined to the MN are being dropped or, in more efficient solutions, buffered at the network and forwarded later to the MN. On the contrary, a soft handover is a *make before break process*, during which the MN is simultaneously connected to both access points. A horizontal handover is a hard handover as only one interface is used, whereas a vertical handover can either be soft or hard based on its design.

Real time applications impose strict limitations on the acceptable end-to-end delay. In Recommendation G.114 [9], the acceptable one-way delay for voice applications is defined to be between 0 and 150 ms. Handover management and especially hard handovers struggle to meet these limitations inducing additional delay to the end-to-end packet delay. This work is focusing on minimising the effect of the inevitable loss of connectivity during hard handovers in VoIP applications. While, the present work is generic and applicable to any mobility protocol, special reference is made below in network layer mobility. This is because IP mobility is the most promising solution towards integrated heterogeneous wireless systems, as network layer is the natural place for these operations [24]. In IP mobility the fast handover concept achieves the best performance in terms of handover delay and packet loss. Fast handover schemes [12, 25] exploit movement detection algorithms to detect that a handover is imminent and identify the new point of attachment to which the mobile node is most likely to move. As a result, network layer operations are initiated while the mobile node is still connected to the current point of attachment, expediting the handover process. Packets destined to the mobile node are buffered at the network side and delivered as soon as the mobile node establishes link-layer connection. Previous work showed that these schemes can achieve acceptable performance regarding user satisfaction [15], when the handover delay is low. However, if handover delay increases, the packets that are buffered during fast handovers would be dropped by the VoIP application due to late arrival. This results to waste of network resources and cancels the benefits of the promising concept of fast handovers.

Fig. 1

MIH event service flow model

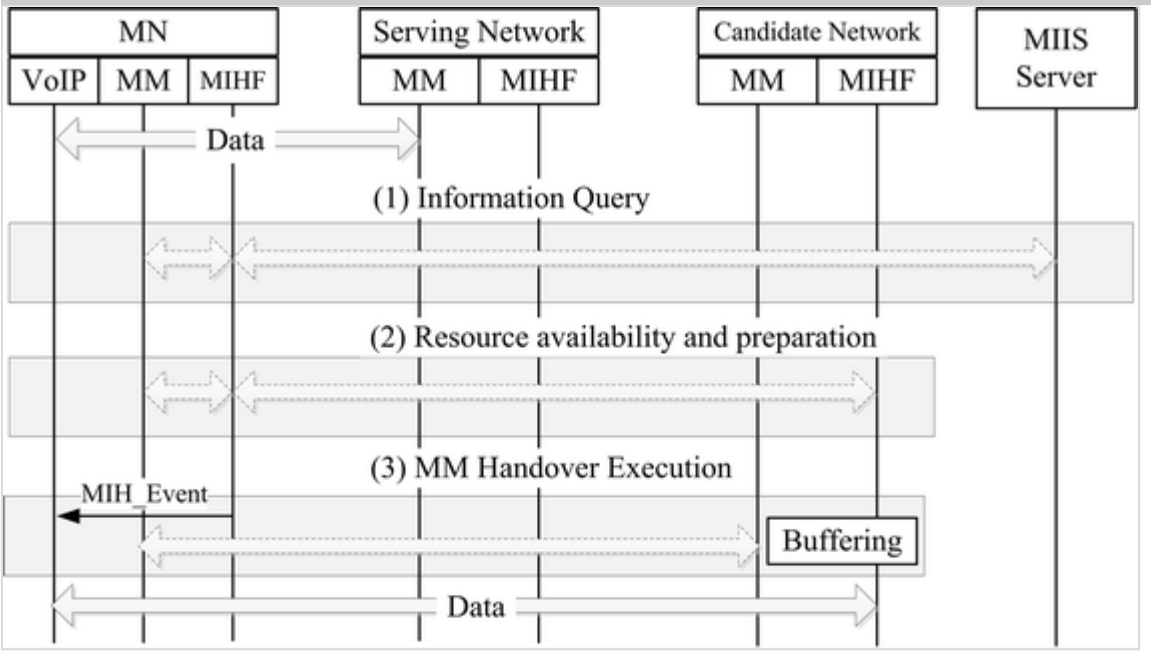

MIH User
MIHF
Link Layer
MIH_Capability_Discover.request
MIH_Capability_Discovery.confirm
MIH_Event_Subscribe.request
Link_Event_Subscribe.request
Link_Event_Subscribe.confirm
MIH_Event_Subscribe.confirm
Event
Link_Event.indication
MIH_Event.indication

MIH framework, also known as IEEE 802.21, is recently introduced in order to provide link-layer intelligence and other related network information to upper layers (i.e., network layer and above) to optimise handovers. MIH can assist the previously discussed fast handovers concept as well as support buffering of packets at network side during handovers by providing link-layer intelligence. The function that realises MIH services is called MIH function (MIHF) and acts as an intermediate layer between link layer and upper layers [8]. Moreover, such link-layer information can be located within the same node or even within a remote node. An upper layer that uses services provided by the MIHF is called MIH user and can generally be a mobility management protocol or an application. The standard defines three types of services that comprise the MIHF. A media independent event service (MIES) that provides event reporting corresponding to dynamic changes in link quality. A media independent command service (MICS) that facilitates the control of the link layer by MIH users. And finally, a media independent information service (MIIS) that provides information of the serving and neighbouring networks.

An event indicates a change in the link layer or predicts an imminent change. The destination of an event is the MIHF (link event) or an MIH user (MIH event). MIH users can subscribe to MIHF in order to receive event notifications of a specific type. As shown in Fig. 1, applications that need access to local link-layer events (i.e., link events of the same protocol stack), they first have to perform MIH capability discovery and then MIH event subscription. The MIH-enabled application generates an MIH Capability Discovery request message and the MIHF responds with an MIH Capability Discovery confirm message. After identifying the available MIH events the application indicates a list of events under interest with an MIH Event Subscribe request. Finally the local MIHF replies with an MIH Event Subscription confirm primitive returning the result of the previous request.

Fig. 2

Overall handover flow chart



MIES supports several categories of link events such as state change events, predictive events, and link handover events. State change events correspond to changes in link layer state. Such events are the MIH Link Down and MIH Link Up events. Predictive events (e.g., MIH Link Going Down) calculate the likelihood of a change in link condition in the near future based on post and present information of the wireless link. Finally link handover events inform upper layers about the occurrence of a link layer handover (e.g., MIH Link Handover Imminent). All the above events can be exploited by an efficient playout scheduling algorithm in order to dynamically adapt to handover delay.

The overall flow chart of the proposed scheme is shown in Fig. 2. Initially, the MN queries the information server to get information about the candidate network (e.g., SSID of 802.11 Access Point). For presentation convenience, all the related signalling which involves several MIH primitives is depicted by a single dashed-outline arrow. Subsequently, the MN queries the availability of the resources at the target network and then reserves them. Then, the MN performs handover to the target network and the MIHF informs the VoIP application about this event. During this phase, the on-the-fly packets are buffered at network side. Finally, the handover process is completed and the MN receives packets through the new path.

3. Playout Algorithms

While mobility management aims to minimise losses due to handover process, playout scheduling tries to tackle packet losses caused by network delay and jitter. Alternatively, when inevitable losses occur playout algorithms exploit concealment methods and redundancies in the source voice stream to mitigate the deteriorating effects. At the sender side, the periodic sampling of a voice source results to a constant packet generation interval with a typical value of 20 ms for most VoIP applications. Afterwards, packets experience random delays while traversing the network. As a result, a playout buffer is introduced at the receiver side, buffering packets for adequate time in order to smooth out such delays and jitter. On the other hand, an increase in packet end-to-end delay degrades the quality of an interactive application such as VoIP. As it can be understood, there is a trade-off between the playout buffer size and the packet loss due to late arrival (late packet loss). An efficient playout algorithm should minimise the late packet loss while applying the minimum buffer delay.

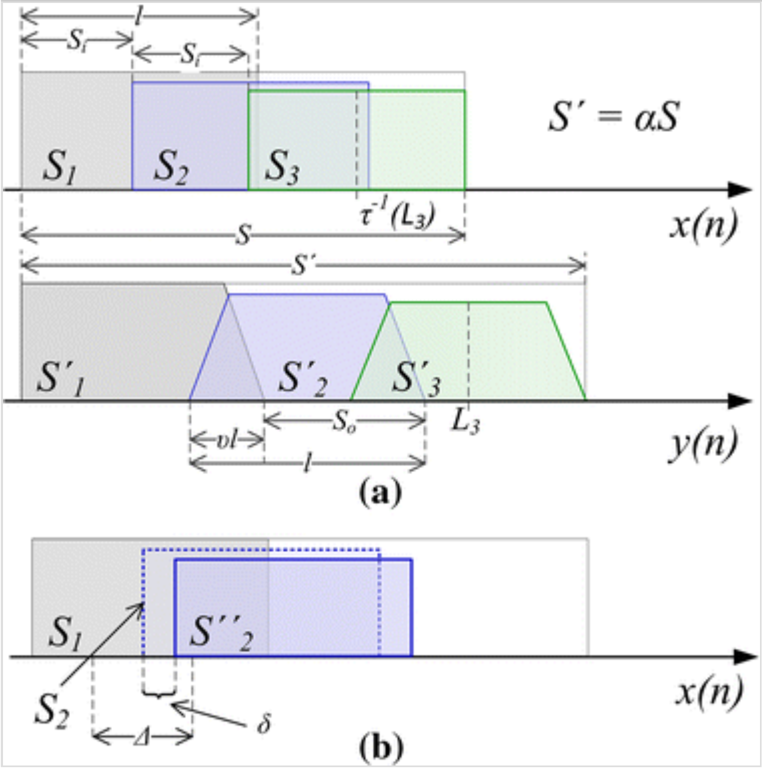
Playout algorithms can be generally categorised into fixed and adaptive. While fixed playout algorithms use the same playout deadline for all packets in a session, adaptive playout algorithms adjust the playout time during silence periods. For that purpose, an estimation of network delay and jitter is used based on previous observations [21]. If the estimated network delay has increased during the last talkspurt, the silence period is extended in order to supply with adequate playout time the next talkspurt, and vice versa. Packet-based adaptation is also possible, through time scaling modifications on voice packets [11, 13]. In this case, expanding a packet results to higher playout delay in order to compensate for an increase in network delay. Inversely, a decrease in the estimated network delay will result to packet compression and shorter playout delay. This approach enables playout adjustment in a highly dynamic way, as well as, faster reactions to varying network conditions given that talkspurt periods might last for some hundred of milliseconds.

Even with adaptive playout scheduling at packet level, a certain amount of packet loss is expected, either as late packet loss or network loss. A variety of loss concealment techniques have been proposed in order to recover the lost signal segments [6, 13]. The simplest concealment method is merely to set all samples of a lost packet to zero and to accept the distortion caused by substituting silence for the missing speech. It is noted that the silence substitution algorithm is not detrimental for low packet loss rates around 1% and, at the same time, the least computationally expensive approach. In particular, results have shown that silence substitution up to 30 ms is tolerable [6].

However, for packet loss rates greater than 1%, approaches that try to reconstruct the waveform of the missing packets are more preferable. Such a concealment method is called waveform substitution [6] and refers to reconstruction of missing packets by substitution of past waveform segments. This method achieves improvement in speech quality relative to silence substitution and it is based on the assumption that the contents of a lost packet will be similar to immediately preceding speech. Nevertheless, by replacing a missing packet with the previous one, distortions will be introduced due to discontinuities at packet edges. This impairment can be reduced by more intelligent waveform selection techniques which exploit the periodicities in voiced speech, as proposed in [6]. This waveform substitution algorithm scans a search window in previous packets to find a region that best matches the one came just before the missing packet. Then, the raised-cosine function is applied to the reconstructed packet and the previous one in order to smooth out the transition. Waveform substitution introduces negligible algorithmic delay due to the pattern matching method applied to a small search window. Finally, another concealment method is based on a hybrid of time-scale modification and waveform substitution [13]. The concealment of a lost packet is accomplished by expanding the last received packet by a factor of 2. When burst losses of consecutive packets occur, waveform repetition is used to conceal an additional lost packet.

Fig. 3

Illustration of **a** OLA algorithm and **b** WSOLA decomposition process



Below, time-scale modification techniques are presented, which are crucial for the current research. The goal of time scale modification algorithms is to modify the rate of speech without affecting the spectral characteristics of speech. A basic method for time scale modification is the overlap-add (OLA) algorithm which is a synthesis operation based on signal interpolation. As shown in Fig. 3, the input signal $x(n)$ is decomposed by extracting k overlapping segments of l samples length which are also regularly spaced with a step of S_i samples. The time-scaled output $y(n)$ by a factor of α is synthesised by overlapping consecutive segments. The overlapping is performed by normalising the sum of these segments with a time-varying normalisation weight. For instance, as depicted in Fig. 3, a trapezoidal window function is applied at segment S_2 to formulate S_2'. Then, S_2' is overlapped and added to the previous reconstructed signal, producing S_o additional samples. For a given overlapping factor of v, the synthesis step S_o is:

$$S_o = (1 - v) \cdot l \qquad 1$$

And the segmentation step for a given set of l, v, and α is:

$$S_i = \frac{(1 - v) \cdot l}{\alpha} \qquad 2$$

Let $\tau(\cdot)$ denotes the time-warp function that does transformation mapping of input signal’s time-basis to output signal’s time-basis (e.g., $L_3 = \tau^{-1}(L_3)$). By choosing synthesis instants to be regularly spaced, such that $L_k = k \cdot S_o$, the corresponding synthesis equation around synthesis instant L_k becomes:

$$y(n) = \frac{\sum_k w^2(n - kS_o) \cdot x(n - kS_o + \tau^{-1}(kS_o))}{\sum_k w^2(n - kS_o)} \qquad 3$$

where $w(\cdot)$ is the window function.

By extracting the segments at fixed intervals, the similarity of the overlapping regions is not taken into consideration. The existing discontinuity between the successive segments and the misaligned interpolation of them result to artefacts and distortions that are detrimental to speech quality. Waveform similarity OLA (WSOLA) [23], on the other hand, searches for a position that has a maximal local similarity (e.g., maximise the cross-correlation function) with the last played out segment before extracting a new segment.

As shown in Fig. 3, instead of extracting segment S_2 that comes exactly S_i samples after S_1, a tolerance region $[-\Delta/2, \Delta/2]$ around $\tau^{-1}(L_1)$ is being searched in order to find the value δ that maximises the similarity measure with respect to the signal portion that follows after S_1. The result is segment S''_2 with length of l samples. A common choice in speech processing for the addition of the successive segments is a hanning window with $v = 0.5$ [23], which results to:

$$\sum_k w^2(n - kS_o) = 1. \qquad 4$$

Then, Equation 3 for the case of WSOLA is rewritten as:

$$y(n) = \sum_k w^2(n - kS_o) \cdot x(n - kS_o + \tau^{-1}(kS_o) + \delta)) \qquad 5$$

It is noted that the search region for maximal local similarity is small compared to the segment size and thus the deviation from the regular segmentation step is negligible. By allowing this tolerance on the decomposition process of the input signal, significant improvement can be observed. At the same time, the WSOLA is computationally and algorithmically more efficient compare to other time-scale modification techniques (e.g., Synchronous OLA or time-domain pitch synchronised OLA) [13, 23].

Typical values for parameters l and Δ are 20 and 5 ms, respectively. These values can be used in the presence of input signal of adequate duration that corresponds to two or three packets. In the case of a single packet, a modified WSOLA algorithm, proposed in [13], can be exploited that uses a smaller segment length (around 10 ms). If the produced output is smaller than the one request, additional iterations are performed on the output to reach the desired length. In both cases, the acceptable values of the scaling factor are within the range of [0.3, 2.0]. Exceeding this range creates audible artefacts and the use of the WSOLA technique is not recommended. Obviously, the single packet WSOLA algorithm can not achieve the same performance with the typical algorithm as it uses less information and searches in a smaller region to find the position with the best similarity.

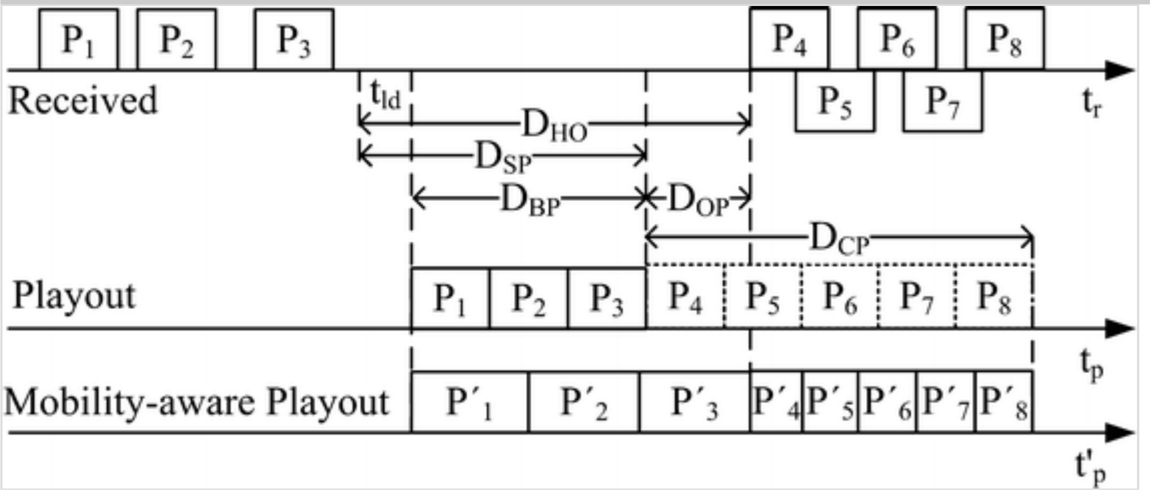
Finally, it is noted that the implementation of the WSOLA technique is both algorithmic and computational efficient [23]. In [20], the authors present an analysis-synthesis system that it is based on the discrete short-time Fourier Transformation. The bulk of the computation in both the analysis and synthesis procedures is performed by the fast Fourier transformation (FFT). In particular performing the computations using the FFT algorithm results to savings of approximately $\log N$ versus N operations per output value. This is based on the Cooley–Tukey algorithm [5], that for an N-point radix-2 FFT of a complex input requires approximately $3N \log N$ real multiplications and $3N \log N$ real additions [16]. Thus the OLA procedure is attractive for implementation as it has a time complexity of $\mathcal{O}(N \log N)$. Moreover, the WSOLA uses an asynchronous segmentation technique with a fixed length window in combination with regularly spaced synthesis intervals, thus it is computationally and algorithmically more efficient than time-scaling modification techniques [23]. Finally, it is noted that, the scaling of voice packets occurs only during the handover period (no more than 200 ms), thus the trade-off between power or processor consumption and the gain in voice quality is considered to be beneficial.

4. Proposed Playout Algorithm

Based on previous discussion, there is an apparent interaction between playout time and handover delay. If the handover delay is higher than the total playout duration of all available packets in the playout buffer, then packets arriving after the completion of the handover process will be late and will be dropped. Both fixed and adaptive playout scheduling algorithms can not react dynamically to this burst delay created by the handover process, dropping the packets that should have been received during the handover. As discussed previously, efficient mobility protocols take special care to buffer these packets at the network side during handovers and then deliver them at the new point of attachment. Taking all the above into consideration, it becomes clear that this lack of synchronisation between the mobility protocol and the voice application leads to waste of network resources and cancels any separate design advantages. This work intends to evaluate the benefits of orchestrating the handover process and the playout schedule using MIH. For that purpose, link-layer awareness is combined with time scaling of voice packets in order to compensate for the handover outage period.

Fig. 4

Conventional and mobility-aware playout timeline during a handover



The following analysis utilises the MIH Link Down event notification and an estimation of the handover delay to calculate the needed scaling of voice packets during and right after the handover process. In order to estimate the handover delay, the playout algorithm can exploit the MIH Link Parameters Report in case that this metric is available for the target network. Alternatively, the mobile node can keep a record of the delay of previous handovers and calculate an expected value. It is assumed that for a specific region and a relatively small time period the handover delay would not vary significantly.

Figure 4 illustrates the timeline of reception, conventional playout, and mobility-aware playout scheduling of packets during and immediately after a handover, denoted as t_r, t_p, and t_p' respectively. Packet P_1 is the first packet that is scheduled to be played out directly after a link down event which takes place at t_{ld}. Packets P_2 and P_3 have also been received and buffered at the client before the occurrence of the link down event. On the contrary, packets P_4 to P_8 are buffered at the network side during the handover and received almost simultaneously after the completion of the handover.

The first important time period in the following analysis is the buffer playout duration, denoted as D_{BP}, which is the total playout duration of the packets present at the playout buffer just before the link down event. Furthermore, D_{HO} stands for the handover delay, which is defined as the time interval between the link breakdown and the moment the mobile node can receive data packets again. In addition, D_{SP} indicates the duration of the supported period, which is the time interval between the link breakdown and the moment the playout buffer is drained (i.e., the moment the last packet was played out). Finally, the outage period, denoted as D_{OP}, is the time interval between the moment the playout buffer was drained and the time the mobile node starts receiving packets again.

Then, the outage period can be calculated as follows:

$$D_{OP} = D_{HO} - D_{SP} = D_{HO} - (t_p^i - t_{dl}) \qquad 6$$

where t_p^i is the already scheduled playout time of packet P_i, which is the first packet to be received after the completion of the handover process. In the example of Fig. 4, $i = 4$. Finally, the scaling factor α for expanding the available packets at the playout buffer is:

$$\alpha \cdot D_{BP} = D_{BP} + D_{OP} \Leftrightarrow \alpha = 1 + \frac{D_{OP}}{D_{BP}} \qquad 7$$

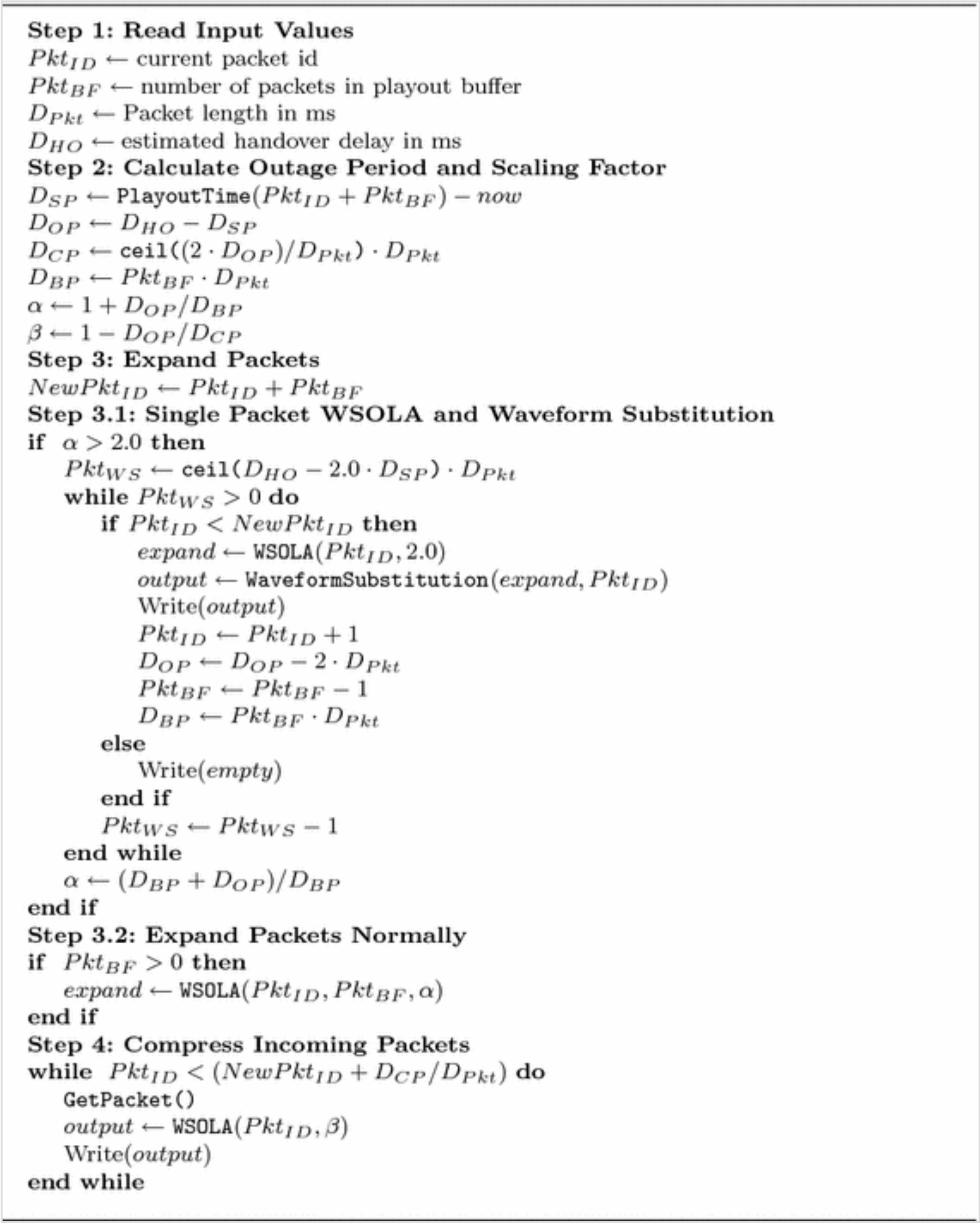
As it is shown in Fig. 4, in the case of conventional playout scheduling (t_p), packets P_4 to P_8 will be dropped (dashed outline), as they arrive after their scheduled playout time. This is a waste of network resources such as buffering memory at the network side, and bandwidth. At the same time, the efficiency of an intelligent handover protocol, that takes special care to buffer packets during handover, is nullified. Such protocols are for instance the fast handovers for mobile IPv6 [12] and fast handovers for proxy mobile IPv6 [25]. On the contrary, with the proposed algorithm these packets are played out after being compressed. The compression scaling factor β can be calculated as follows:

$$\beta \cdot D_{CP} = D_{CP} - D_{OP} \Leftrightarrow \beta = 1 - \frac{D_{OP}}{D_{CP}} \qquad 8$$

where D_{CP}, is the duration of the compression period which is the total initial playout out duration of the packets that are going to be compressed. In the example of Fig. 4, D_{CP} is the playout duration of packets P_4 to P_8. After D_{CP} elapses, the upcoming packets will be played out normally (e.g., packet P_9 in the previous example). If the D_{CP} is too small, the compression will be too high resulting to artefacts. On the other hand, if D_{CP} is too high, more packets are going to be compressed. Typical values of factor β should be in the range of 0.4 to 0.7 but not lower than 0.3. Alternatively, for a given value of D_{OP}, D_{CP} can be extracted by setting a desirable value of β.

Fig. 5

Mobility-aware adaptive playout algorithm



```
Step 1: Read Input Values
$Pkt_{ID} \leftarrow$ current packet id
$Pkt_{BF} \leftarrow$ number of packets in playout buffer
$D_{Pkt} \leftarrow$ Packet length in ms
$D_{HO} \leftarrow$ estimated handover delay in ms
Step 2: Calculate Outage Period and Scaling Factor
$D_{SP} \leftarrow \texttt{PlayoutTime}(Pkt_{ID} + Pkt_{BF}) - now$
$D_{OP} \leftarrow D_{HO} - D_{SP}$
$D_{CP} \leftarrow \texttt{ceil}((2 \cdot D_{OP})/D_{Pkt}) \cdot D_{Pkt}$
$D_{BP} \leftarrow Pkt_{BF} \cdot D_{Pkt}$
$\alpha \leftarrow 1 + D_{OP}/D_{BP}$
$\beta \leftarrow 1 - D_{OP}/D_{CP}$
Step 3: Expand Packets
$NewPkt_{ID} \leftarrow Pkt_{ID} + Pkt_{BF}$
Step 3.1: Single Packet WSOLA and Waveform Substitution
if $\alpha > 2.0$ then
    $Pkt_{WS} \leftarrow \texttt{ceil}(D_{HO} - 2.0 \cdot D_{SP}) \cdot D_{Pkt}$
    while $Pkt_{WS} > 0$ do
        if $Pkt_{ID} < NewPkt_{ID}$ then
            $expand \leftarrow \texttt{WSOLA}(Pkt_{ID}, 2.0)$
            $output \leftarrow \texttt{WaveformSubstitution}(expand, Pkt_{ID})$
            Write($output$)
            $Pkt_{ID} \leftarrow Pkt_{ID} + 1$
            $D_{OP} \leftarrow D_{OP} - 2 \cdot D_{Pkt}$
            $Pkt_{BF} \leftarrow Pkt_{BF} - 1$
            $D_{BP} \leftarrow Pkt_{BF} \cdot D_{Pkt}$
        else
            Write($empty$)
        end if
        $Pkt_{WS} \leftarrow Pkt_{WS} - 1$
    end while
    $\alpha \leftarrow (D_{BP} + D_{OP})/D_{BP}$
end if
Step 3.2: Expand Packets Normally
if $Pkt_{BF} > 0$ then
    $expand \leftarrow \texttt{WSOLA}(Pkt_{ID}, Pkt_{BF}, \alpha)$
end if
Step 4: Compress Incoming Packets
while $Pkt_{ID} < (NewPkt_{ID} + D_{CP}/D_{Pkt})$ do
    GetPacket()
    $output \leftarrow \texttt{WSOLA}(Pkt_{ID}, \beta)$
    Write($output$)
end while
```

Additionally, when $D_{HO} > 2.0 \cdot D_{BP}$, the handover delay will exceed the maximum available expanded signal playout duration, as the maximum expansion factor should not be greater than 2. In this case a similar approach to [13] can be followed. Firstly, every packet in the playout buffer is expanded to the maximum length (i.e., doubled) and then further expanded through waveform substitution. Our proposed adaptation algorithm is presented in Fig. 5. Upon the reception of an MIH Link Down event, the proposed playout algorithm is triggered. Initially, the scaling factor α is calculated. If α is higher than 2, the single packet WSOLA method is executed for as many packets as needed. Otherwise, the typical WSOLA algorithm is performed having as input signal all the available packets at the playout buffer. If the handover exceeds the available reconstruction period of $3 \cdot D_{BP}$ then silence substitution is performed. As a result link-layer intelligence utilises the single packet WSOLA only when the regular algorithm is not sufficient and thus achieving a better performance than [13].

Finally, it is also noted that expansion of available packets can start before the MIH Link Down event, by utilising the MIH Link Going Down or the MIH Link Handover Imminent events. This approach can be more applicable when the playout buffer occupancy is low and the maximum scaling of available packets after the MIH Link Down event might not be sufficient to bridge the gap created by the handover process. However, after the MIH Link Down event has occurred the previous analysis still stands to calculate the expansion of the following packets.

5. Subjective Listening Test Results

As described in Sect. 3, fixed and adaptive playout algorithms can only exploit concealment methods to recover burst packet losses, such those introduced by the handover process. As a result, the proposed algorithm performance was evaluated against silence substitution, waveform substitution and WSOLA concealment methods. To do so, a simulation program was developed which is capable of distorting voice files in a similar manner that a handover would. Afterwards, subjective listening tests were conducted in order to obtain the MOS of each scheme.

5.1. Simulation Environment

Fig. 6

Sequence diagram of simulator

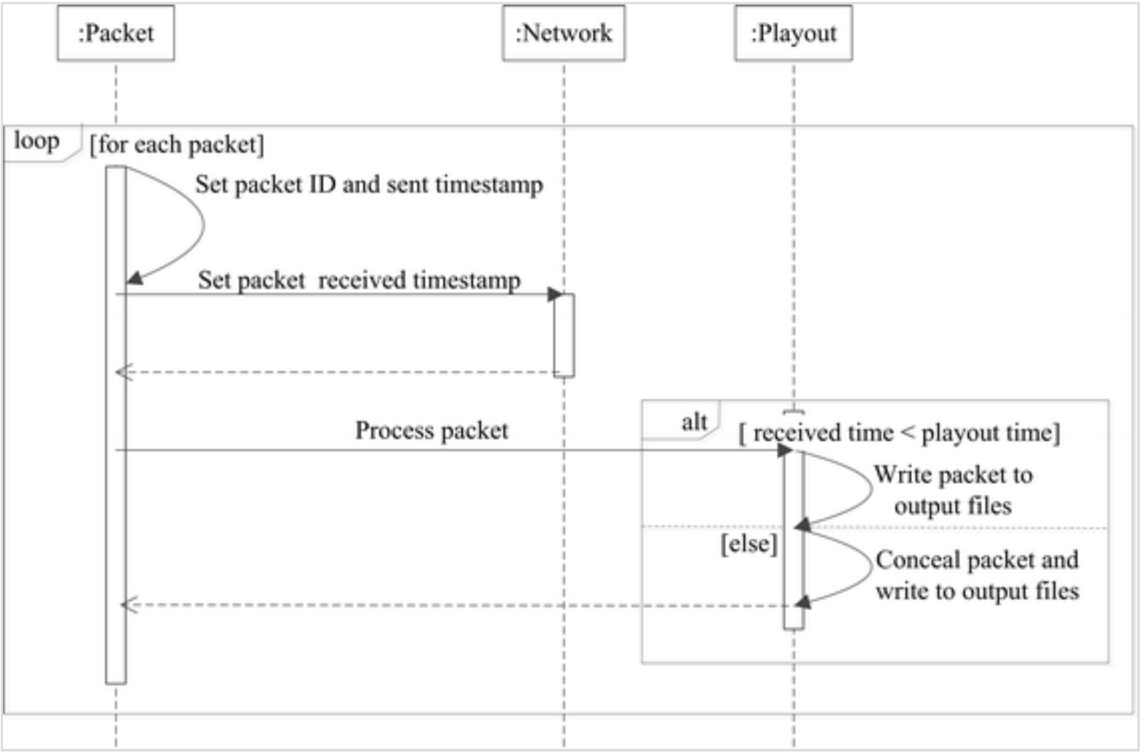

:Packet
:Network
:Playout
loop
[for each packet]
Set packet ID and sent timestamp
Set packet received timestamp
Process packet
alt
[received time < playout time]
Write packet to output files
[else]
Conceal packet and write to output files

Table 1

Simulation parameters

Parameter	Value(s)
Network delay	50 ms
Handover delay	120, 160, 200 ms
Audio sample rate	8 kHz
Packet size	160 samples, 20 ms
Window function	Hanning
Segment length	20 ms
Overlapping factor	0.5
Similarity measure	Cross-correlation
Search region	5 ms

The simulation program was implemented in C++ language in order to introduce distortion to voice files based on the scenarios under test. The arguments passed to the program are a voice file, the time the handover starts, and the duration of the handover. The generated output is one distorted file for each investigated scheme. Figure 6 depicts a generic sequence diagram of the simulator. As it can be seen, the program is composed by the following classes:

1. Packet class: The objects of this class represent packets with variables the packet identification number, the time the packet was generated at the source, the time the packet was received at the destination, and the scheduled playout time. Each packet in the C++ domain corresponds to specific samples of the input voice file. As a result, when a packet is played out in the simulator the corresponding samples are written at the output file.

2. Network class: During simulation, the network class assigns values to the generation and reception time variables of each packet. The applied network delay can be constant or stochastic following a specific distribution. For the packets involved in the handover the class calculates and applies the additional delay. As the main focus of this work is the late packet loss caused by the handover process, the network delay was fixed to 50 ms throughout the simulation. So the only losses observed are due to handover process.

3. Playout class: This is the most essential class of the program, as it simulates the behaviour of a playout scheduling algorithm. Initially, the size of the playout buffer is calculated. The fixed network delay of 50 ms results to a fixed D_{BP} of 60 ms. If a packet was received before its playout time, then the corresponding samples of the packet are written to the output voice file. In the case of late packet loss, the playout class conceals the lost packet as described in Sect. 3. A different output file is created for each one of the four methods in a single simulation run. For the waveform substitution algorithm the cross-correlation function was used with a search window of 20 ms. This class also implements the WSOLA algorithm as described in Sect. 3. The corresponding WSOLA() method takes as arguments the identifiers of the input packets and the scaling factor and returns the synthesised output signal. Similar to [13, 23], the WSOLA algorithm uses a hanning window function, segment length of 20 ms, overlapping factor of 0.5, cross-correlation function as the similarity measure and a search region of 5 ms. All these parameters are summarized in Table 1.

Fig. 7

Original and simulator generated voice files

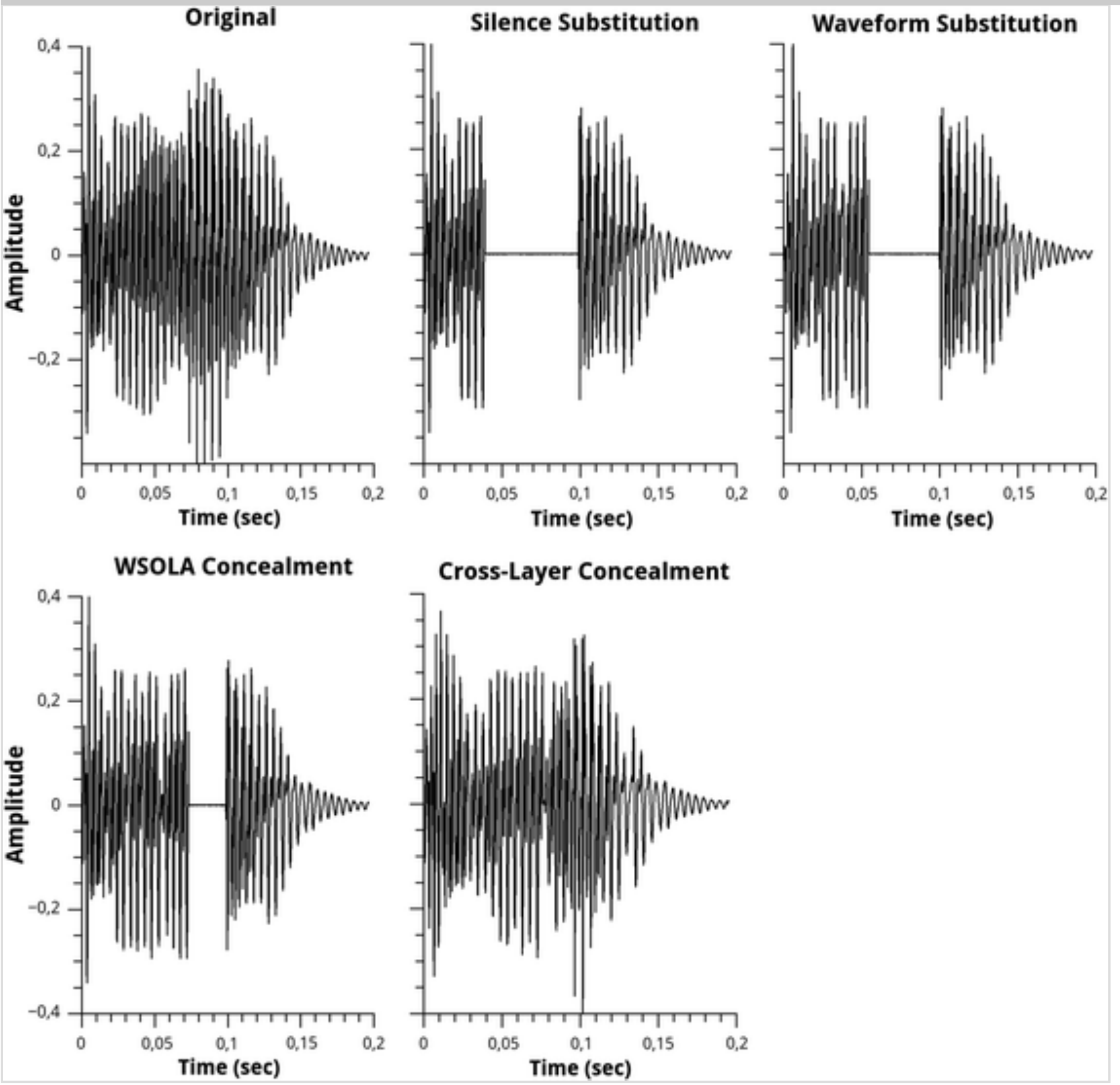


Figure 7 demonstrates the waveform of the original recording as well as the ones obtained by the simulator for all cases. As it can be seen, the silence substitution is not reconstructing any portion of the lost data and simply replaces the lost portion with silence of equal duration. Waveform substitution can reconstruct missing parts of the signal by substituting them with past waveform segments. However the reconstructed portion of the lost samples is up to double the length of the last received packet. The remaining part is substituted with silence. The WSOLA algorithm further improves the concealment as it exploits time-scaling modifications and can double the last packets length. However, this means that only the last received packet can be exploited by WSOLA in order to conceal the outage period. Finally, our proposed mobility-aware cross-layer playout algorithm exhibits the best performance because it acts ahead of the handover, based on the provided link-layer intelligence, utilising all packets available at the playout buffer. Thus, our solution can scale all available packets and reconstruct all the missing signal.

5.2. Subjective Listening Tests

The speech material consists of simple, meaningful, short, and unrelated sentences, recorded by four, male and female, speakers. Each sample is composed of two sentences separated by 0.5 s of silence. For comparison purposes, this work used, similarly to previous studies [2], well-known codecs. Unless stated otherwise, the reference voice files were initially encoded using G.711 Alaw codec with 64 kbps bit rate and 8 kHz sample rate. Each packet carries 160 samples which results to 20 ms of playout time. Eighteen listeners participated in the test, both male and female of various age groups. The score for each signal processing condition is obtained by averaging the scores from all listeners and samples. All above parameters are summarized in Table 1.

To evaluate the investigated algorithms the degradation category rating (DCR) method was used as described in Annex D of ITU-T Recommendation P.800 [10]. As the handover process lasts only for some hundreds of milliseconds, the impairment is small and the DCR method is more suitable as it affords higher sensitivity. The DCR method compares the system under test with a high quality fixed reference in the format of “reference sample - distorted sample” to allow higher sensitivity by direct comparison of pairs. Afterwards, the degradation is rated on a five-point scale corresponding to 5-inaudible, 4-audible but not annoying, 3-slightly annoying, 2-annoying, and 1-very annoying, respectively. The scores obtained in this way are referred to as degradation MOS (DMOS). Also reference conditions were included in which the original sample is played twice in order to anchor the judgement of the listeners. The null pair of the reference conditions received a score of 4.9, which assures the validity of the testing methodology.

5.3. Results

5.3.1. Impact of Handover Delay

Fig. 8

DMOS versus handover delay

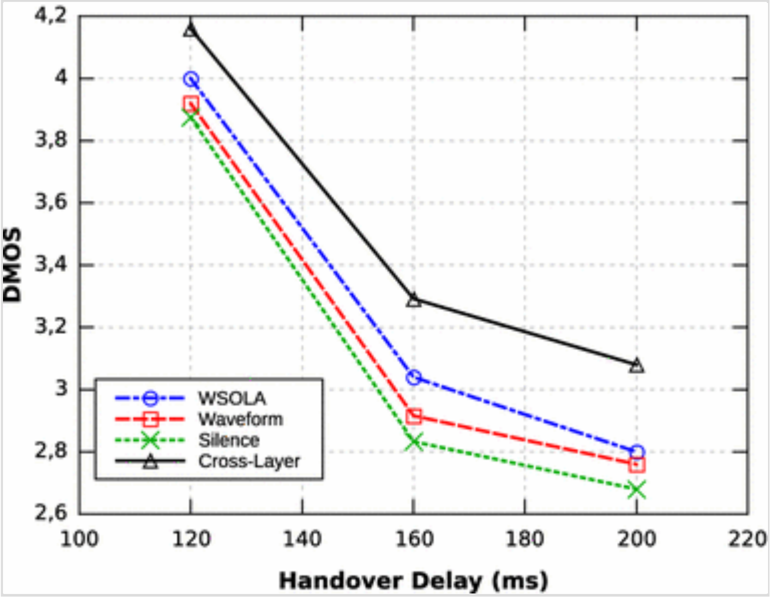


Figure 8 illustrates the obtained DMOS values as a function of the handover delay, D_{HO}, for all schemes under test. As already mentioned, the ITU considers one-way delay between 0 and 150 ms as acceptable for voice applications in Recommendation G.114 [9]. Moreover, in our previous work on handover delay [15], we have discovered that the degradation of speech quality is more severe in the region between 120 and 200 ms. As we want to validate our algorithm under stretched conditions, we have used the same interval. As it can be seen, the proposed mobility-aware playout scheduling algorithm denoted as “Cross - Layer” considerably outperforms the proposed methods of silence substitution (“Silence”), waveform substitution (“Waveform”), and WSOLA concealment (“WSOLA”). Moreover, Fig. 8 shows that the initial slope for all algorithms is steeper for handover values between 120 and 160 ms compared to 160 and 200 ms. This is in accordance to previous findings, showing that low packet loss rate is more tolerable.

Among the algorithms under test, the silence substitution exhibits the worst performance (i.e., DMOS values from 3.88 at 120 ms to 2.68 at 200 ms) as it is not reconstructing any portion of the lost data. A slightly better performance is achieved by waveform substitution (i.e., 3.92 at

120 ms to 2.76 at 200 ms), however with out time scaling modifications the reconstructed portion of the lost samples is too small. The “WSOLA” algorithm further improves the speech quality achieving values from 4.03 at 120 ms to 2.81 at 200 ms. However, this algorithm awaits for the first packet to be lost in order to conceal losses and to adapt the playout scheduling. This means that only the last received packet can be exploited by WSOLA in order to conceal the outage period. Finally, the proposed mobility-aware playout algorithm exhibits the best performance with DMOS values from 4.16 at 120 ms to 3.08 at 200 ms. This is because the algorithm reacts in advance, based on the provided link-layer intelligence, utilising all packets available at the playout buffer. As a result, the proposed scheme achieves a gain of at least 0.3 DMOS points compare to other concealment methods even at the extreme cause of $D_{HO} = 200$ ms.

5.3.2. Impact of Buffer Playout Duration

Fig. 9

DMOS for different D_{BP} values

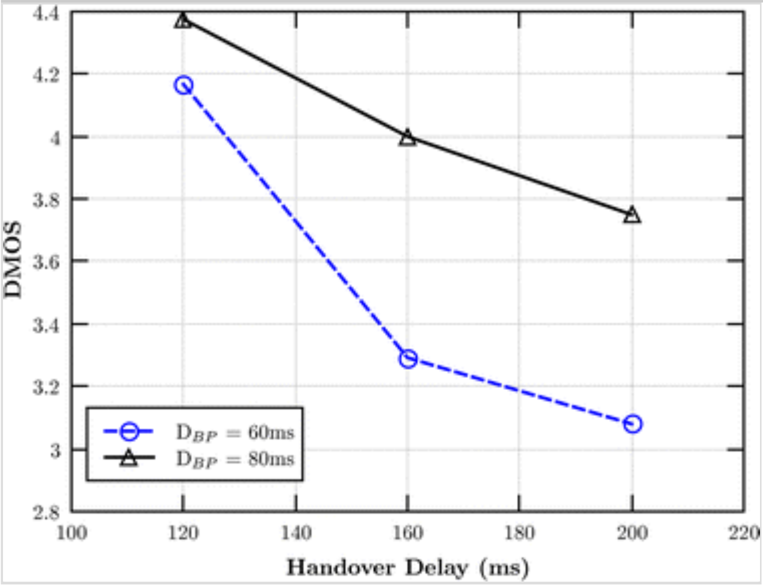


Figure 9 depicts the obtained DMOS values of the proposed scheme as a function of the handover delay for different values of the buffer playout duration D_{BP}. Additionally to the previous presented results for $D_{BP} = 60$ ms, supplementary results were obtained for $D_{BP} = 80$ ms.

This means that an additional packet is available at the playout buffer just before the handover. As it is shown, the gain in DMOS value varies from 0.2 at $D_{HO} = 120$ ms to 0.7 $D_{HO} = 200$ ms, further highlighting the capabilities of the proposed algorithm. This observation is a result of the two folded impact of the additional packet at the playout buffer. Firstly, the outage period is further reduced by 20 ms while, secondly, a lower scaling factor is applied to the involved packets.

5.3.3. Impact of Codec Compression

Fig. 10

DMOS for different codecs

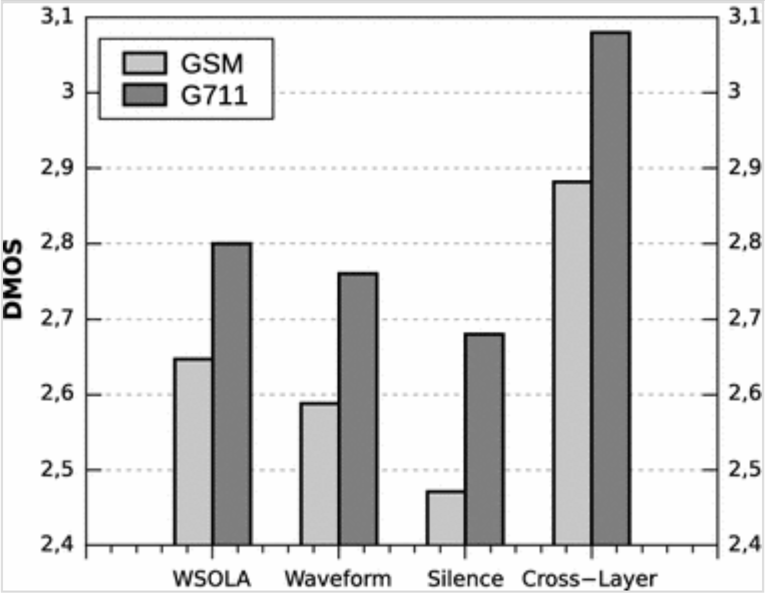


Figure 10 displays the impact of using a codec with a higher compression ratio on all evaluated scenarios. For this case the full-rate GSM codec was used, which transforms blocks of 160 speech samples that span 20 ms resulting to 8000 Hz sample rate and 13 kbps data rate. For this case the handover delay was set to 200 ms and the buffer playout duration to 60 ms. While in the case of GSM codec a degradation in speech quality is observed compared to G.711 codec for all scenarios, the proposed scheme is the least affected. In particular, the decrease in DMOS value for the proposed scheme is 0.08 while for the other cases varies between 0.15 and 0.2. This remark is justified by the fact that the degradation of speech quality due to higher compression ratio in conjunction with packet loss is significantly detrimental to voice quality. On the contrary, the reconstructed signal of the proposed scheme conceals the packet loss and achieves better overall performance.

6. Conclusion

In this paper, the combination of the MIH protocol with the WSOLA time-scale modification technique was investigated for packet-based adaptive playout scheduling for VoIP applications. The proposed scheme does not introduce any modification at the network and the sender side and only explores the advantages of the recently proposed MIH framework which adds link-layer intelligence to the Internet Protocol stack. As a result the proposed scheme is only implemented at the application layer of the receiver and thus this work is widely applicable. Results indicated that in order to realise the benefits of link-layer intelligence at the mobility management stack, optimisations at the application layer are also needed. Specifically, the proposed playout scheduling algorithm can better utilise the buffering techniques of mobility management protocols, conserving network resources and achieving better quality of experience. In particular, if the playout scheduling adapts to handover process, a gain of 0.3 DMOS points is achieved compared to previous proposed playout scheduling algorithms and concealment techniques. The importance of the proposed scheme is further emphasised by the increased frequency of handovers that is anticipated in future heterogeneous wireless environments.

References

1. Al-Surmi, I., Othman, M., & Ali, B. M. (2012). Mobility management for IP-based next generation mobile networks: Review, challenge and perspective. *Journal of Network and Computer Applications*, *35*(1), 295–315.

2. Bernaschi, M., Cacace, F., Iannello, G., & Vellucci, M. (2007). Mobility management for VoIP on heterogeneous networks: Evaluation of adaptive schemes. *IEEE Transactions on Mobile Computing*, *6*(9), 1035–1047.

3. Carneiro, G., Ruela, J., & Ricardo, M. (2004). Cross-layer design in 4G wireless terminals. *IEEE Wireless Communications*, *11*(2), 7–13.

4. Choi, H. H., Song, O., Park, Y. K., & Lee, J. R. (2010). Performance evaluation of opportunistic vertical handover considering on–off characteristics of VoIP traffic. *IEEE Transactions on Vehicular Technology*, *59*(6), 3115–3121.

5. Cooley, J., & Tukey, J. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, *19*(90), 297–301.

6. Goodman, D., Lockhart, G., Wasem, O., & Wong, W. C. (1986). Waveform substitution techniques for recovering missing speech segments in packet voice communications. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *34*(6), 1440–1448.

7. Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., & Patil, B. (2008). *Proxy Mobile IPv6*. IETF RFC 5213.

8. IEEE LAN/MAN Std 802.21-2008. (2010). *IEEE standard for local and metropolitan area networks*. Part 21: Media Independent Handover Service.

9. ITU-T Recommendation G.114. (2003). *One-way transmission time*.

10. ITU-T Recommendation P. 800. (1996). *Methods for subjective determination of transmission quality*.

11. Kim, H. G., & Lee, J. H. (2012). Enhancing VoIP speech quality using combined playout control and signal reconstruction. *IEEE Transactions on Consumer Electronics*, *58*(2), 562–569.

12. Koodli, R. (2009). *Fast handovers for mobile IPv6*. IETF RFC 5568.

13. Liang, Y., Farber, N., & Girod, B. (2003). Adaptive playout

scheduling and loss concealment for voice communication over IP networks. *IEEE Transactions on Multimedia*, *5*(4), 532–543.

14. Lykourgiotis, A., & Kotsopoulos, S. (2010) Comparison and evaluation of the most efficient mobility management protocols for integrated wireless networks. In *10th International conference on next generation wired/wireless networking (NEW2AN'10)* (pp. 377–388). Berlin: Springer.

15. Lykourgiotis, A., Kotsopoulos, S., & Dagiuklas, T. (2012). A comprehensive simulation study of low latency handoffs in mobile IPv4 for VoIP in IEEE 802.11 b WLAN. In *15th ACM MSWiM poster session and proceedings of the 7th ACM PM2HW2N workshop* (pp. 29–36). ACM.

16. Mitra, S., & Kaiser, J. (1993). *Handbook for digital signal processing*. New York: Wiley.

17. Mohanty, S., & Akyildiz, I. (2006). A cross-layer (layer 2 + 3) handoff management protocol for next-generation wireless systems. *IEEE Transactions on Mobile Computing*, *5*(10), 1347–1360.

18. Nasser, N., Hasswa, A., & Hassanein, H. (2006). Handoffs in fourth generation heterogeneous networks. *IEEE Communications Magazine*, *44*(10), 96–103.

19. Perkins, C., Johnson, D., & Arkko, J. (2011). *Mobility support in IPv6*. IETF RFC 6275.

20. Portnoff, M. (1976). Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *24*(3), 243–248.

21. Ramjee, R., Kurose, J., Towsley, D., & Schulzrinne, H. (1994). Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *IEEE conference on computer communications (INFOCOM'94)* (pp. 680–688). IEEE.

22. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,

Peterson, J., Sparks, R., et al. (2002). *SIP: Session Initiation Protocol*. IETF RFC 3261.

23. Verhelst, W., & Roelands, M. (1993). An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In *1993 IEEE international conference on acoustics, speech, and signal processing (ICASSP'93)* (vol. 2, pp. 554–557). IEEE.

24. Wozniak, J. (2015). Mobility management solutions for current IP and future networks. *Telecommunication Systems*, 1–19.

AQ2

25. Yokota, H., Chowdhury, K., Koodli, R., Patil, B., & Xia, F. (2010). *Fast handovers for proxy mobile IPv6*. IETF RFC 5949.