

# A synchronization control scheme for videoconferencing services

Ivano Bartoli<sup>1</sup>, Giovanni Iacovoni<sup>2</sup>, Fabio Ubaldi<sup>1</sup>

<sup>1</sup>Co.Ri.TeL., Rome, Italy

Email: {ivano.bartoli, fabio.ubaldi}@coritel.it

<sup>2</sup>University of Rome "La Sapienza", Rome, Italy

Email: iacovoni@die.uniroma1.it

**Abstract**—We propose a synchronization control scheme which achieves both speech/video Intra-Stream synchronizations and Inter-Stream synchronization for videoconferencing services over IP networks. The driving principle of our scheme is to guarantee the Intra-Sync speech timing relationships (hence the speech quality) and to adjust the video Intra-Sync and the Inter-Sync accordingly. Towards this aim we use a preventive control for the speech stream and a reactive control for the video stream. More precisely, we use an adaptive playout algorithm that keeps the Intra-Sync constraints within the talkspurts, while the network jitters are compensated by modifying only the silence period lengths on the basis of both speech and video packet delays. We implemented our scheme in a prototype, which allowed us to test the effectiveness of our solution. Actually, we could appreciate both perfect speech intelligibility and very satisfactory user perceived lip-sync. The latter is because the Inter-Sync error is concentrated only at the beginning of the talkspurts, where known experimental tests have shown that it is not detectable.

**Index Terms**—Lip synchronization, Intra synchronization, talkspurt-based adaptive playout, videoconferencing service

## I. INTRODUCTION

In video conferencing applications speech and video Media Data Units (MDUs) must keep the following temporal relationships at the recipient's side:

- 1) A temporal relationship within each stream called "Intra-Stream synchronization" (for example a video stream captured at 10 fps implies that each video MDU has to be displayed for 100ms);
- 2) A temporal relationship between speech and video streams called "Inter-Stream synchronization" or "lip-sync" (speech and video MDUs captured at the same time have to be played out together).

These are hard tasks to be accomplished in the Internet. First of all the connectionless nature of the Internet introduces variable delays on each stream, so altering the Intra-Stream synchronization of both speech and video streams. Besides, speech and video MDUs are periodically captured and sent through the network by using two different sessions. As a result also the Inter-Stream synchronization may be altered at the endpoint, degrading the user perceived quality: a classical effect is

the utterance of the speaker without the corresponding movements of the lips.

Although there are different solutions proposed in literature, the current videoconferencing products (both commercial and freeware) do not implement any synchronization control scheme. The main reason is that generally in an IP videoconference the skew between speech and video hardly exceeds 80 ms, which is the minimum skew detectable by a human subject [1]. However in the last few years, with the growth of IP networks, the scenario has changed and the lip-sync problem is getting more and more crucial: for instance, a videoconferencing session between two users, each one connected to a different network, requires signal processing on the media stream. That processing could introduce different delays between the two streams (for example a video transcoding process requires more computational time than speech transcoding).

So a synchronization control scheme is needed to check and recover the lost synchronization. It must be designed by taking into account:

- Trade-off between Inter-Stream and Intra-Stream synchronization. Generally, the recovery of the Inter-Stream synchronization may cause a change in the Intra-Stream synchronization and viceversa.
- Trade-off between end-to-end delay and packet loss. The synchronization schemes require the use of buffering technique adding a certain amount of delay in the communication. Since in IP videoconferencing scenarios the maximum communication delay recommended by ITU is 400 ms [2], a synchronization scheme has to meet this constraint and keep the buffer size as small as possible. On the other hand, if the buffer size is too small, the MDU loss rate is likely to increase; as a consequence, the quality experienced by the user is degraded. For these reasons all synchronization policies have to find a good compromise in order to provide an acceptable user perceived quality.

Each type of synchronization error has a different impact on the user perception: it is well-known that

user perception is less affected by the video Intra-Stream synchronization than by the speech Intra-Stream synchronization, where even errors of few milliseconds are detected and considered annoying.

The different impact of the speech and of the video Intra-Stream synchronization on user perception calls for the adoption of the master/slave strategy. The latter was proposed for the first time in [3]: the speech stream is the "master" and determines the MDUs playout time by keeping its Intra-Stream synchronization error as low as possible; the video stream acts as a "slave" and passively adjusts its MDU playout time to achieve the lip-sync and, when it is possible, the video Intra-Stream synchronization. Different algorithms implement the above-mentioned master/slave strategy. The two most relevant solutions are described in [4] and [5]. The solution proposed in [4] adopts the master/slave *switching strategy*: each stream computes its MDUs buffering times according to the Intra-Stream constraints. For each speech/video couple of MDUs to be played out simultaneously, the stream with lower buffering time becomes the slave and increases the delay in order to achieve the lip-sync. The master/slave strategy proposed in [5] is an improvement of the implementation proposed in [6], and it has been further refined in [7] and applied to Wi-Fi technology in [8]. Briefly, [5] defines a set of thresholds related to the loss ratio and to the Intra-Sync error for both speech and video streams. Their synchronization control scheme tries to keep these two metrics under the thresholds by accordingly increasing (and for the speech also decreasing) the end-to-end delay. Even if these techniques allow to perfectly track the ever changing network conditions, their drawback is that such a continuous reaction may result in annoying artifacts when listening to the speech content. Actually, short "gaps" within the talkspurt could be very annoying.

Based on this observation we propose a synchronization control scheme based on both preventive and reactive<sup>1</sup> techniques.

The preventive part is based on the well-known adaptive playout algorithm [9] that adapts to network jitter by modifying only the length of the silence periods. This guarantees a very low speech Intra-Sync error during the talkspurt, while obtaining at the same a good trade-off between end-to-end delay and loss ratio.

In our scheme we modified the algorithm [9] by inserting a reactive feature which is in charge of taking into account as much as possible the Inter-Stream constraints. In particular we compute the speech playout times also on the basis of the delays experienced by the video stream, provided that the playout times do not violate the speech Intra-Stream synchronization. As a result the Inter-Sync error happens to be concentrated at the beginning of each talkspurt, where users hardly perceive it [10].

<sup>1</sup>Preventive control techniques try to avoid loss of synchronization while reactive control techniques recover from loss of synchronization after it has occurred

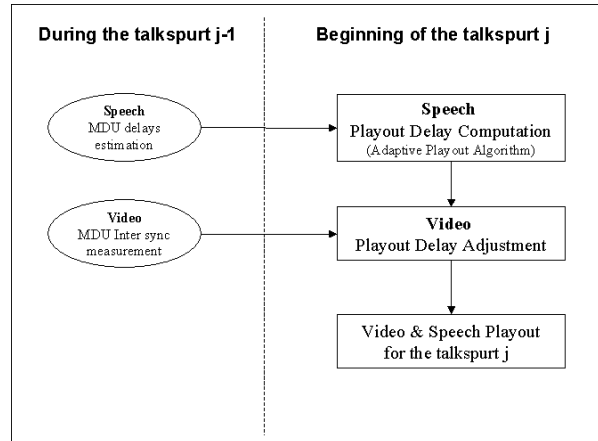


Figure 1. Flowchart of the proposed synchronization scheme

We implemented our integrated solution in a videoconferencing test-bed, with a suitable delay emulator placed between the two videoconferencing end-points. We tested our scheme under a variety of different delay patterns for both the speech and the video streams. We could observe live the effectiveness of our scheme since the speech is perfectly intelligible and at the same time a certain lip-sync level is guaranteed.

However, since we set up an accurate test environment, we focused also on objective measurements, by comparing the performance of our scheme with those obtained by Liu's scheme<sup>2</sup> described in [5]. The comparisons were made by looking at the usual Intra and Inter Stream metrics, as well by using the E-model framework for the speech quality evaluation. Objective metric results suggest that our scheme improves the speech quality, while the Inter-Sync is almost the same. The video Intra-Sync is the less accurate, but it is balanced by a reduced loss ratio; besides it is at the same time the less relevant from a user perspective's viewpoint.

The paper is organized as follows: in Section II we detail our synchronization control scheme; in Section III we describe the testing methodology for our proposed scheme; we discuss the obtained results in Section IV. Section V concludes the paper.

## II. LIP SYNCHRONIZATION CONTROL SCHEME

### A. Basic ideas

The flowchart in Figure 1 shows the top-level behaviour of our solution. At the beginning of each talkspurt (i.e. when the first speech MDU is received after a silence period), the adaptive playout algorithm computes the "playout delay" for the talkspurt, i.e. the amount of time from when the MDU is generated by the source until it is played out at the destination.

<sup>2</sup>To the best of our knowledge this is the most recent one which includes clear and reproducible network conditions and measurement results

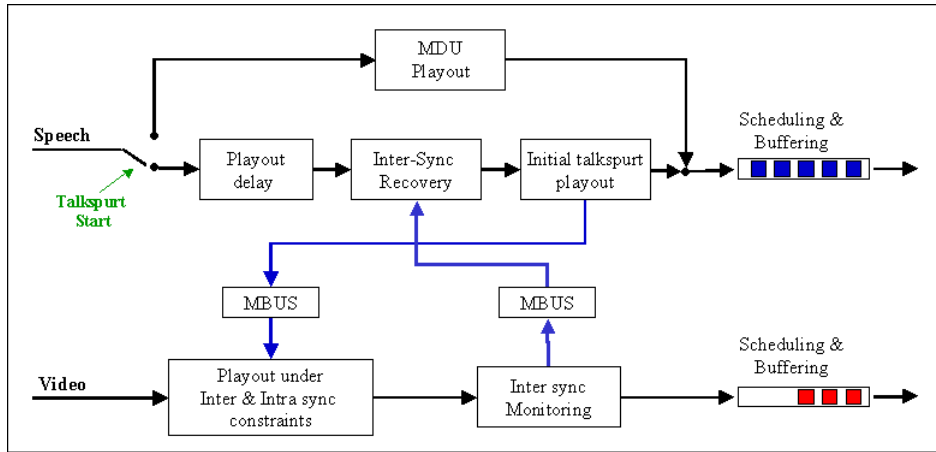


Figure 2. Lip-Synchronization Scheme

Afterwards, this playout delay can be increased on the basis of a number of video Inter-Sync measurements collected up to the last received video MDU. The aim of this adjustment is to take into account an extra amount of delay so as to fulfill the Inter-Sync constraints. We remark that this behaviour doesn't mean that a master/slave switching occurs, since the adjustment is carried out only if the speech intra sync constraints are still met. Then, speech and video playout times are accordingly computed. As a further remark, we stress that the output duration of every speech MDU is left unchanged. A common source reference clock for speech and video is required. In our implementation we rely on the RTP/RTCP protocols and therefore we are able to compute the timing relationship between speech and video by means of the RTP timestamps and the RTCP Sender Report (RTCP-SR) informations (see Section III-A).

```

for each speech MDU  $i$  {
  if (MDU  $i$  is the first unit of talkspurt  $j$ ) {
     $B_p(j) = \text{Playout\_delay};$ 
     $B_p(j) = \text{Inter\_sync\_recovery}(B_p(j), B_p(j-1), T_{err});$ 
     $t_{talk}^p(j) = \text{Initial\_talkspurt\_playout}(B_p(j), t_{talk}^g(j));$ 
    send ( $t_{talk}^p(j), t_{talk}^g(j)$ );
  } else {
     $t_s^p(i) = \text{MDU\_payout}(B_p(j));$ 
  }
}
    
```

Figure 3. Speech stream handling pseudo-code. Section II-B.1 explains the symbols

Symbol	Meaning
$t_s^g(i)$	Generation time of speech MDU $i$
$t_s^p(i)$	Playout time of speech MDU $i$
$B_p(j)$	"playout delay" for talkspurt $j$
$t_{talk}^p(j)$	Playout time of the first speech MDU of talkspurt $j$
$t_{talk}^g(j)$	Generation time of the first speech MDU of talkspurt $j$
$t_v^g(i)$	Generation time of video MDU $i$
$t_v^{tp}(i)$	Target playout time of video MDU $i$
$t_v^p(i)$	Playout time of video MDU $i$
$t_v^{arr}(i)$	Arrival time of video MDU $i$
$T_{err}$	Estimated Inter-Synchronization error

TABLE I.  
LIST OF SYMBOLS

**B. Proposed solution details**

In this subsection we describe our scheme. For the sake of clarity, speech stream handling and video stream handling are described separately by referring to Figure 2 and by using the notations in Table I.

1) *Speech stream handling*: The speech stream is handled at the receiver side according to the pseudo-code shown in Figure 3. The scheme performs one of

the following tasks after having checked the content of the MDU, since the latter can be either or not the first unit of the  $j^{th}$  talkspurt<sup>3</sup>:

- MDU  $i$  is the first unit of the  $j^{th}$  talkspurt. The following three steps are carried out, one after the other:
  - *Playout\_delay*. The adaptive playout algorithm (implemented in the RAT tool and described in [9]) computes the "playout delay"  $B_p(j)$  for talkspurt  $j$ , which takes into account a suitable buffering time based on the network delay estimated up to the last received MDU of the previous talkspurt<sup>4</sup>.
  - *Inter\_sync\_recovery*. At this step, the current video Inter-Sync measurement  $T_{err}$  (see subsection II-B.2) is taken into account by adjusting the playout delay  $B_p(j)$  as follows:

$$B_p(j) = \max\{B_p(j-1) + T_{err}, B_p(j)\} \quad (1)$$

<sup>3</sup>In our implementation the beginning of a talkspurt is explicitly marked in the RTP header

<sup>4</sup> $B_p(j) = \hat{d} + \beta\hat{\sigma}$ , where  $\beta = 3$ ,  $\hat{d}$  is the estimated average delay and  $\hat{\sigma}$  the estimated delay standard deviation.

- *Initial\_talkspurt\_playout*. The value  $t_{talk}^p(j)$  is computed according to the playout delay  $B_p(j)$ :

$$t_{talk}^p(j) = t_{talk}^g(j) + B_p(j) \quad (2)$$

The playout time  $t_{talk}^p(j)$  and the corresponding generation time  $t_{talk}^g(j)$  represent the time reference couple which is also used by the Video stream to compute its playout times during the talkspurt  $j$ .

- MDU  $i$  is not the first unit of the  $j^{th}$  talkspurt. Then MDU playout time  $t_s^p(i)$  is computed in order to keep the intra synchronization within the talkspurt  $j$ , i.e. the same playout delay  $B_p(j)$  is used to compute playout times of all the MDUs belonging to the talkspurt:

$$t_s^p(i) = t_s^g(i) + B_p(j) \quad (3)$$

2) *Video stream handling*: The two blocks in the lower part of Figure 2 are here explained.

```

for each video MDU  $i$  {
   $t_v^{tp}(i) = MDU\_playout(t_{talk}^p(j-1), t_{talk}^g(j-1), t_v^g(i));$ 
  schedule MDU  $i$ ;
  if (new reference  $(t_{talk}^p(j), t_{talk}^g(j))$ ) {
    Check_scheduler $(t_{talk}^p(j), t_{talk}^g(j));$ 
  }
}

Check_scheduler $(t_{talk}^p(j), t_{talk}^g(j))$  {
  if  $(t_{talk}^g(j) < t_v^g(k))$  {
    for  $i = k$  to  $k + N - 1$  {
       $t_v^{tp}(i) = MDU\_playout(t_{talk}^p(j), t_{talk}^g(j), t_v^g(i));$ 
    }
  } else if  $(t_v^g(k) \leq t_{talk}^g(j) \leq t_v^g(k + N - 1))$  {
    for  $i = k$  to  $k + N - 1$  {
      if  $(t_v^g(i) \geq t_{talk}^g(j))$ 
         $t_v^{tp}(i) = MDU\_playout(t_{talk}^p(j), t_{talk}^g(j), t_v^g(i));$ 
    }
  }
}

```

Figure 4. Video stream handling pseudo-code. Section II-B.2 explains the symbols

- *Playout under INTER & INTRA constraints* (see pseudo-code in Figure 4). For each Video MDU  $i$ , the corresponding playout time  $t_v^{tp}(i)$  is computed according to both Inter-Sync and Intra-Sync constraints as follows:

- *MDU\_playout*. In this step, we compute the “target playout” for the MDU  $i$ . It is the playout time that meets the Inter-Sync constraints.

$$t_v^{tp}(i) = t_{talk}^p(j-1) + [t_v^g(i) - t_{talk}^g(j-1)] \quad (4)$$

where  $(t_{talk}^p(j-1), t_{talk}^g(j-1))$  is the reference couple of the speech. So the MDU is scheduled to be played out at the target playout time  $t_v^{tp}(i)$ .

If the video stream receives a new reference couple  $(t_{talk}^p(j), t_{talk}^g(j))$  an additional step is carried out:

- *Check\_scheduler*. Target playout times of MDUs inside the scheduling are adjusted according to the new reference. Specifically, suppose that the scheduler contains  $N$  MDUs  $(k, \dots, k+N-1)$ , with generation times  $t_v^g(i)$ ,  $i = k \dots k+N-1$ . Besides, suppose that the MDUs are ordered according to their generation time, i.e.  $t_v^g(k) \leq t_v^g(k+1) \leq \dots \leq t_v^g(k+N-1)$ . Three possible situations could occur:
  - \*  $t_{talk}^g(j) < t_v^g(k)$  In such a situation the target playout times of all MDUs inside the scheduler will be adjusted using Equation (4) and the new reference couple.
  - \*  $t_v^g(k) \leq t_{talk}^g(j) \leq t_v^g(k+N-1)$ . In such a situation only the MDUs with generation times greater than  $t_{talk}^g(j)$  will be adjusted.
  - \*  $t_{talk}^g(j) > t_v^g(k+N-1)$ . The playout time of MDUs inside the scheduler are unchanged.

It's easy to demonstrate that Equation (4) satisfies the Intra-Sync relationship between two consecutive video MDUs  $i$  and  $i+1$ , when the target playout times and are computed using the same reference couple  $(t_{talk}^p(j), t_{talk}^g(j))$ . Otherwise an Intra-Sync error is introduced in order to meet the Inter-Sync constraints by shortening/extending the output duration of video MDUs.

Every time a reference is changed by the speech stream a transient period is required to reach the Inter-Synchronization. Since this changes happens at the beginning of the talkspurts, the lip-sync error is concentrated in such points. This behaviour is desired because in [10] it is shown that a short period of lip asynchrony at the beginning of an utterance followed by a complete lip sync for the rest of the talkspurt is not perceived by a user (at least up to a speech/video skew of 300 ms).

- *Inter-Sync Monitoring*. Due to the network conditions, video MDUs could be received out-of-time in order to be played out at target playout time. Then, for each MDU  $i$  the assigned target playout time  $t_v^{tp}(i)$  is matched with the effective measured playout time  $t_v^p(i)$  and a value  $S_{err}(i)$  is computed as:

$$S_{err}(i) = t_v^p(i) - t_v^{tp}(i) \quad (5)$$

We use a fixed window that continuously collects the  $S_{err}$  of the last  $W$  received MDU. If there are less than  $l$  positive  $S_{err}$  they are removed since they are considered outliers. Then,  $T_{err}$  is computed as:

$$T_{err} = \max(S_{err}(i)); \quad (6)$$

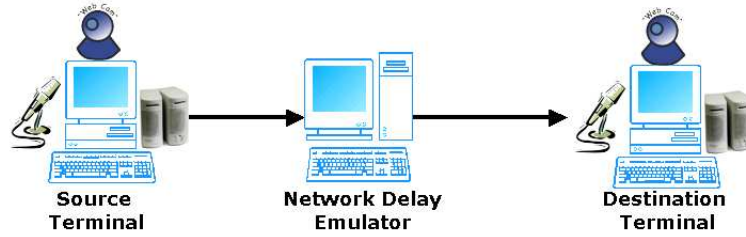


Figure 5. Testbed Scheme

If  $T_{err}$  is greater than zero, an Inter-Sync error is detected and sent to the speech stream, otherwise the video stream allows for an overall playback delay reduction.

### III. TESTING METHODOLOGY

#### A. Test Environment

In order to evaluate the performance of the solution, we designed and implemented a test-bed. It is described in sections III-A (Test Environment), III-B (Network Delay) and III-C (Objective performance evaluation).

We tested our synchronization scheme in the test environment depicted in Figure 5, which consists of two Linux-based terminals connected through a network delay emulator<sup>5</sup>. Both terminals are equipped with open-source videoconferencing tools VIC [12] and RAT [13], wherein we implemented our scheme.

Since these tools rely on the RTP/RTCP protocols, the timing relationships between speech and video MDUs can be reconstructed at the receiver side using the “timestamp” field in the RTP packet header of both speech and video streams, which gives the generation time of the MDU in RTP units. Besides, we used the “RTP timestamp” and “NTP timestamp” fields in the RTCP-SR packets, i.e. the generation time of the Sender Report in RTP and NTP units respectively. These two fields are used to identify a common source reference clock for the two streams as explained in the following.

Let us consider a speech MDU with RTP timestamp  $T_s^{RTP}$  and a video MDU with RTP timestamp  $T_v^{RTP}$ . Let  $(T_s^{RTP-SR}, T_s^{NTP-SR})$  and  $(T_v^{RTP-SR}, T_v^{NTP-SR})$  be the RTP/NTP timestamps carried in the speech and video RTCP-SR respectively. The difference  $\Delta T$  between the generation times of the MDUs can be easily computed as follows:

- 1) Convert the speech RTP timestamp to NTP units:  

$$T_s^{NTP} = T_s^{NTP-SR} + (T_s^{RTP} - T_s^{RTP-SR}) * \Phi_{RTP \rightarrow NTP}^s$$
- 2) Convert the video RTP timestamp to NTP units:  

$$T_v^{NTP} = T_v^{NTP-SR} + (T_v^{RTP} - T_v^{RTP-SR}) * \Phi_{RTP \rightarrow NTP}^v$$
- 3) Compute the difference as:  

$$\Delta T = (T_s^{NTP} - T_v^{NTP}) * \Phi_{NTP \rightarrow ms}$$

<sup>5</sup>We used the Network Simulator 2 [11] “network emulation capability”

where  $\Phi_{RTP \rightarrow NTP}^s$  and  $\Phi_{RTP \rightarrow NTP}^v$  are the conversion factors from RTP units to NTP for speech and video respectively, while  $\Phi_{NTP \rightarrow ms}$  is the conversion factor from NTP units to milliseconds.

The communication between RAT and VIC is carried out by means of the Local Message Bus (MBUS) protocol [14]. Last but not least, we removed the clock skew between the two sound cards [15], since it affects the network delay estimates of a generic adaptive playout. That removal is done by using an algorithm based on the minimum delay method which works as follows. For each  $K$  delay measures (we set  $K = 100$ ) the minimum value is collected: this is the measurement which is affected the least by the skew. Afterwards the so collected minimum values are fitted using the least-square method in order to obtain the slope of the skew, enabling us to correct accordingly the measured values.

#### B. Network Delay

Whenever possible we compared the performance of our scheme with the one proposed in [5], therefore using the same network delay model and parameter values. Specifically, we emulated three network behaviors, which will be called “Moderate”, “Bad” and “Severe” in the rest of the paper. For the first two behaviours we generated series of pseudo-network delay values using a two-state Markov model. For each state  $i$  ( $i = 0, 1$ ) the series  $\{d_i\}$  are generated according to the Left-Truncated Gaussian Distribution, defined as:

$$p(d_i) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(d_i - \mu_i)}{2\sigma_i^2}} & , d_i \geq m_i \\ 0 & , d_i < m_i \end{cases} \quad (7)$$

where  $\mu_i$ ,  $\sigma_i$  and  $m_i$  are the mean, the standard deviation and the truncation value respectively.

The parameters used to generate the Moderate and the Bad behaviors are shown in Table II, where  $p_{0 \rightarrow 1}$  and  $p_{1 \rightarrow 0}$  are the transition probabilities for the Markov model. The “Severe” behaviour, i.e. the situation of abrupt changes in network condition, is obtained by switching between Moderate and Bad behaviours.

#### C. Objective performance evaluation

Since the test-bed described in III-A is able to provide a live real-time service, we appreciated the effectiveness

	$\mu_0$	$\sigma_0$	$m_0$	$\mu_1$	$\sigma_1$	$m_1$	$p_{0 \rightarrow 1}$	$p_{1 \rightarrow 0}$
Moderate	50 ms	10 ms	40 ms	75 ms	10 ms	65 ms	0.01	0.1
Bad	100 ms	50 ms	50 ms	180 ms	70 ms	110 ms	0.01	0.04

TABLE II.  
NETWORK DELAY PARAMETERS

Video		Speech	
Codec	H.263+	Codec	AMR-WB
Bit rate	64 Kbit/s	Bit rate	23.6 Kbit/s
Frame rate	15 fps	Frame duration	20 ms
Format	QCIF		

TABLE III.  
VIDEO AND SPEECH PARAMETERS

of our solution just by looking at the media received at the Destination Terminal. However, in order to provide also a set of objective measure results, we used a test sequence captured off-line to achieve a reasonable statistics reliability target. In fact, by doing so we could run the test several times on the same test sequence but with different realizations of the same delay distribution. Unfortunately, the sequence used in [5] is not available. Then, we resorted to a sequence of a speaker in head view in a TV news environment and we adopted, whenever possible, the same coding parameters (see Table III). The sequence lasts 10 minutes and the speech and the video streams of this sequence are the source of the videoconferencing session.

As far as the performance evaluation are concerned, we used the following set of metrics, where  $N_s$  ( $N_v$ ) is the number of MDUs of the speech (video) stream:

- Inter-Sync RMSE<sup>6</sup> ( $\delta^{inter}$ ).  

$$\delta^{inter} = \sqrt{\frac{\sum_{n=2}^{N_s} [(t_s^p(m) - t_v^p(n)) - (t_s^g(m) - t_v^g(n))]^2}{N_s - 1}}$$
 where  $m$  represent the speech MDU corresponding to the video MDU  $n$ .
- Intra-Sync RMSE ( $\delta^{intra}$ ).  

$$\delta^{intra} = \sqrt{\frac{\sum_{n=2}^{N_i} [(t_i^p(n) - t_i^p(n-1)) - (t_i^g(n) - t_i^g(n-1))]^2}{N_i - 1}}$$
- Loss ratio ( $l_i$ ).
- End-to-end Delay ( $D_i^{tot}$ ). It is the time interval between the MDU generation and its playout time. In order to perform a simple measure of this metric, the system clocks are synchronized via the Network Time Protocol (NTP).

#### IV. EXPERIMENTAL RESULTS

Since our test-bed works on-line, the overall scheme performance are affected by the latencies introduced by the operating system of the Destination Terminal. They are:

- 1) “Cushion size”. In order to counteract the delay introduced by the operating system of the Destination Terminal, the RAT tool [16] estimates a “cushion

size”, i.e. a lower bound for the playout delay so as to prevent sound card buffer from emptying.

- 2) “Operating system losses”. RAT may not process timely incoming packets even if they arrived on time to be played out according to the current scheduling. Therefore, such packets will be discarded.

Since we want to evaluate the performance of our synchronization scheme by itself, in the results presented in this section we removed the delay introduced by the “cushion size” and the losses due to the “operating system losses”.

We generated two sets of different delay series. Each set corresponds to a different network delay scenario described in the following:

- In the first set of tests (Section IV-A), we directly applied to the RTP packets the delay values obtained by the model described in Section III-B.
- In the second set of tests (Section IV-B), we checked the performance of our scheme when the video stream experiences a fixed extra delay with respect to the speech stream (this might correspond to the case of a videoconference where a transcoder is introduced in the communication for bit-rate adaptation).

With reference to the definitions given in section II-B.2, for the monitoring window size  $W$  we chose the value of 5, while the threshold  $l$  was set to 2. Actually, according to our experience, this choice is a good trade-off between a reliable estimate and the need of a fast adaptation to the changing network condition.

Each test is performed by sending the test sequence for each network behaviour (Moderate, Bad, Severe) 100 times from the Source Terminal to the Destination Terminal, each time using a different realization of the delay series. The test results reflect the 95% confidence intervals.

##### A. Basic network delays

Here results are compared with those in [5]. Their master/slave strategy defines a set of thresholds for speech and video performance (maximum loss ratio and maximum Intra-Stream synchronization RMSE). For each

<sup>6</sup>Root Mean Square Error

Algorithm	Ch. behaviour	$\delta_s^{intra}$ [ms]	$l_s$	$D_s^{tot}$ [ms]
PROPOSED	Moderate	$0.10 \pm 0.001$	$1.1 \cdot 10^{-4} \pm 1.5 \cdot 10^{-5}$	$104.16 \pm 0.15$
	Bad	$0.24 \pm 0.04$	$2 \cdot 10^{-2} \pm 5 \cdot 10^{-4}$	$269.0 \pm 0.27$
	Severe	$0.21 \pm 0.01$	$1 \cdot 10^{-2} \pm 3.5 \cdot 10^{-4}$	N/A
LIU'S	Moderate	$1.1 \pm 0.5$	$3.0 \cdot 10^{-4} \pm 5 \cdot 10^{-4}$	$85.8 \pm 6.4$
	Bad	$1.6 \pm 0.3$	$9.6 \cdot 10^{-3} \pm 4.0 \cdot 10^{-3}$	$289.0 \pm 13.2$
	Severe	$2.5 \pm 0.2$	$8.0 \cdot 10^{-3} \pm 2.0 \cdot 10^{-3}$	N/A

TABLE IV.  
COMPARISON OF SPEECH PERFORMANCE UNDER BASIC NETWORK DELAYS

speech/video MDU, these metrics are monitored and if one of them exceeds the threshold the end-to-end delay is increased. On the other hand if the speech metrics don't exceed the thresholds for a fixed monitoring window, the end-to-end delay is decreased.

Table IV compares the performance concerning the speech stream<sup>7</sup>: the adaptive playout strategy results in very low ( $< 0.3ms$ ) Inter-Sync RMSE, regardless of the network behaviours. Such good performance can be achieved without a serious impact on the speech quality in terms of loss ratio and end-to-end delay.

Moreover, we evaluated the speech Mean Opinion Score (MOS) according to the E-model [2] for both Liu's scheme and our proposed scheme; loss and delay impairment factors have been computed using the equations derived by [17] for different codecs: AMR narrowband at the higher bit-rate (14.2 Kb/s) and lower one (4.75 Kb/s), G.723.1, G.729, iLBC (Internet Low Bit-rate Codec). In Table V the performance are compared.

Codec	Liu		Proposed scheme	
	Moderate	Bad	Moderate	Bad
AMR (High)	3.86	2.83	3.85	2.76
AMR (Low)	3.07	2.12	3.06	2.18
G.729	3.53	2.53	3.52	2.54
G.723.1	3.38	2.39	3.37	2.43
iLBC	3.63	2.7	3.62	2.78

TABLE V.  
COMPARISON BETWEEN E-MODEL MOS VALUES

For the Moderate behaviour both schemes have approximately the same performance. For the Bad behaviour, our scheme has a lower end-to-end delay and a higher loss ratio. The MOS is strictly dependent on the type of codec. Then if the codec is more sensitive to the loss impairments than to the delays, the MOS is lower (AMR-High and G.729); otherwise, higher MOS values can be achieved (AMR-Low and iLBC). As far as the Inter-Sync RMSE, the lack of sync is essentially concentrated in short time periods at the beginning of each talkspurt (Figure 6) where it is hardly perceived by a user (see the thorough subjective evaluation carried out in [10]). Therefore we removed the errors at the beginning of the

<sup>7</sup>The average delay in the Severe behaviour is not meaningful due to the non-stationary behaviour

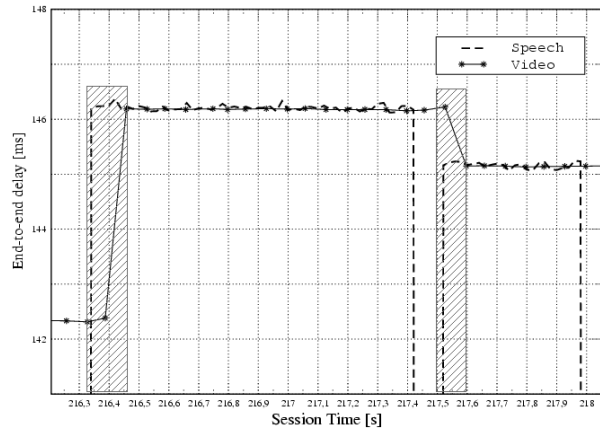


Figure 6. "out-of-sync" (shaded regions) at the beginning of a talkspurt

talkspurts, resulting in a filtered measure shown in Table VI.

Algorithm	Ch. behaviour	$\delta_s^{inter}$ [ms]	$\Delta_t^e$ [ms]
PROPOSED	Moderate	$0.34 \pm 0.02$	$44.73 \pm 0.55$
	Bad	$4.00 \pm 0.1$	$35.15 \pm 1.72$
	Severe	$2.26 \pm 0.1$	$48.15 \pm 0.5$
LIU'S	Moderate	$1.4 \pm 0.6$	—
	Bad	$2.2 \pm 0.5$	—
	Severe	$2.6 \pm 0.9$	—

TABLE VI.  
COMPARISON OF INTER-STREAM RMSE UNDER BASIC NETWORK DELAYS

The same Table VI also shows the average "out-of-sync" duration of the filtered errors ( $\Delta_t^e$ ). The latter are clearly almost negligible: they do not exceed 50 ms, i.e. less than a video frame duration for a video displayed at 15 fps.

As expected, the video Intra-Sync error has the most degraded performance of our scheme: see Table VII, especially when the speech playout delay is quite variable for each talkspurt (Bad and Severe behaviors). However this numerical results must be considered with care. Actually, there are not suitable models to evaluate the video quality degradation due to Intra-Sync alteration. At any rate, since the time duration of a single frame is 67 ms (see table III), in the worst case (Bad network behaviour) the error never exceeds 20%. Besides, [18] shows that the instantaneous video frame rate may affect the speech

Algorithm	Ch. behaviour	$\delta_v^{intra}$ [ms]	$l_v$	$D_v^{tot}$ [ms]
PROPOSED	Moderate	$1.25 \pm 0.02$	$3.8 \cdot 10^{-5} \pm 10^{-5}$	$104.21 \pm 0.15$
	Bad	$14.83 \pm 0.16$	$2.3 \cdot 10^{-2} \pm 7.8 \cdot 10^{-4}$	$271.1 \pm 0.27$
	Severe	$12.63 \pm 0.17$	$9.1 \cdot 10^{-3} \pm 3.9 \cdot 10^{-4}$	N\A
LIU'S	Moderate	$1.3 \pm 0.6$	$4.0 \cdot 10^{-4} \pm 3.0 \cdot 10^{-4}$	$85.8 \pm 6.4$
	Bad	$1.9 \pm 0.5$	$9.6 \cdot 10^{-3} \pm 4.0 \cdot 10^{-3}$	$289.0 \pm 13.3$
	Severe	$3.1 \pm 0.9$	$8.0 \cdot 10^{-3} \pm 4.0 \cdot 10^{-3}$	N\A

TABLE VII.  
COMPARISON OF VIDEO PERFORMANCE UNDER BASIC NETWORK DELAYS

Ch. behaviour	$\delta_s^{intra}$ [ms]	$l_s$	$D_s^{tot}$ [ms]
Moderate	$0.10 \pm 0.001$	$9 \cdot 10^{-5} \pm 1.3 \cdot 10^{-5}$	$147.8 \pm 0.18$
Bad	$0.15 \pm 0.001$	$1.6 \cdot 10^{-3} \pm 1.7 \cdot 10^{-4}$	$305.9 \pm 0.14$
Severe	$0.15 \pm 0.002$	$1.1 \cdot 10^{-2} \pm 2.3 \cdot 10^{-4}$	N\A

TABLE VIII.  
SPEECH PERFORMANCE UNDER SHIFTED NETWORK DELAYS

perceived quality only if it is below 5 fps, which is not our case, and which is in general very unlikely to occur.

### B. Shifted network delays

The synchronization control scheme has been inserted in a videoconference session where it is present a video/speech transcoder that adapts the bit-rate between the two terminals. In this scenario the video stream experiences an higher delay than the speech, due to the larger processing time of video transcoding operations. Specifically, we experimentally evaluated that the video delays are shifted by a length of time approximately constant and equal to  $40ms$ . In each network behaviour (Moderate, Bad, Severe) we added this value to the video delay series.

In Table VIII we observe that the Intra-Synchronization of speech is kept, preserving a good compromise between loss ratio and end-to-end delay.

Ch. behaviour	$\delta_s^{inter}$ [ms]	$\delta_f^{inter}$ [ms]	$\Delta_t^\epsilon$ [ms]
Moderate	$3.34 \pm 0.06$	$3.09 \pm 0.06$	$55.92 \pm 1.47$
Bad	$30.55 \pm 0.3$	$6.95 \pm 0.09$	$92.29 \pm 2.44$
Severe	$25.77 \pm 0.29$	$2.77 \pm 0.15$	$85.06 \pm 0.61$

TABLE IX.  
INTER-STREAM RMSE UNDER SHIFTED NETWORK DELAYS

Table IX shows the Inter-Sync performance. The Intra-Sync RMSE is higher than that in the previous tests and it can be explained as follows. Most of the time the video MDUs experience larger delays than the speech and therefore often arrive out-of-sync. The speech stream reacts to the video communications only at the beginning of the talkspurt resulting in an error, which however is kept quite low. Actually it never exceeds  $80ms$ , which is the threshold recommended by the observations in [1].

At the same time the performances of the video stream are preserved and are about the same as in the previous tests. (see Table X).

Ch. behaviour	$\delta_v^{intra}$ [ms]	$l_v$	$D_v^{tot}$ [ms]
Moderate	$3.40 \pm 0.05$	$9 \cdot 10^{-5} \pm 10^{-5}$	$147.9 \pm 0.15$
Bad	$24.71 \pm 0.19$	$9 \cdot 10^{-5} \pm 10^{-5}$	$303.4 \pm 0.14$
Severe	$17.6 \pm 0.18$	$4.5 \cdot 10^{-4} \pm 10^{-5}$	N\A

TABLE X.  
VIDEO PERFORMANCE UNDER SHIFTED NETWORK DELAYS

## V. CONCLUSION

In this paper we presented a lip synchronization control scheme able to take advantage from both reactive and preventive techniques. The speech Intra-Sync error is kept low by taking advantage of the adaptive playout algorithm. At the same time it is able to guarantee a certain level of lip-sync since we let the video measurements to play a role in the playout time computations. The unavoidable Inter-Sync errors are concentrated almost only at the beginning of the talkspurts, so to have a negligible impact on the user's perception. Finally, the introduced error in the video Intra-Sync doesn't impact the overall perceived quality.

We implemented the lip sync control scheme in a real time test bed which allowed us to perform a wide range of objective measures which gave evidence on the above findings.

## REFERENCES

- [1] R. Steinmetz, "Human Perception of Jitter and Media Synchronization," *IEEE Journal on Selected Area on Communication*, vol. 14, no. 1, Jan. 1996.
- [2] "The E-model, a computational model for use in transmission planning," Recommendation G.107, ITU-T, Mar. 2003.
- [3] K. Rothermel and T. Helbig, "An Adaptive Stream Synchronization Protocol," in *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, Apr. 1995.
- [4] I. Kouvelas, V. Hardmann, and A. Watson, "Lip Synchronization for use over Internet: Analysis and Implementation," in *Proceedings of IEEE Globecom '96*, Nov. 1996.

- [5] H. Liu and M. El Zarki, "A synchronization control scheme for real-time streaming multimedia applications," in *Proceedings of 13th Packet Video Workshop 2003*, Apr. 2003.
- [6] Y. Xie, C. Liu, M. J. Lee, T. Saadawi, and N. Saadawi, "Adaptive Multimedia Synchronization in a Teleconference System," *Multimedia Systems*, vol. 7, no. 4, July 1999.
- [7] H. Liu and M. El Zarki, "On the adaptive delay and Synchronization Control of Video Conferencing over the Internet," in *Proceedings of International Conference on Networking 2004*, Feb. 2004.
- [8] —, "Adaptive delay and synchronization control for Wi-Fi based AV conferencing," in *Proceedings of the First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks 2004*, Oct. 2004.
- [9] R. Ramjee, J. Kurose, and D. Towsley, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks," in *Proceedings of IEEE INFOCOM 1994*, June 1994.
- [10] M. Chen, "Low-latency lip-synchronized videoconferencing system," in *Proceedings of the Conference on Human factors in computing systems*, Apr. 2003.
- [11] "Network simulator 2," <http://nstram.isi.edu/nstram/index.php/>.
- [12] "Vic, videoconferencing tool," <https://frostie.cs.ucl.ac.uk/mbone/mmedia/wiki/VicWiki>.
- [13] "Rat, robust audio tool," <https://frostie.cs.ucl.ac.uk/mbone/mmedia/wiki/RatWiki>.
- [14] K. D. Ott J., Perkins C., "Message Bus For Local Coordination," RFC 3259, Internet Engineering Task Force, Apr. 2002.
- [15] M. R. Delco, "Production Quality Internet Television," Berkeley Multimedia Research Center, Technical Report TR161, Aug. 2001.
- [16] I. Kouvelas and V. Hardman, "Overcoming Workstation Scheduling Problems in a Real-Time Audio Tool," in *Proceedings of the USENIX Annual Technical Conference*, Jan. 1997.
- [17] R. G. Cole and J. Rosenbluth, "Voice over IP Performance Monitoring," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2, Apr. 2001.
- [18] J. Mullin, L. Smallwood, A. Watson, and G. Wilson, "New techniques for assessing audio and video quality in real-time interactive communication," in *Proceedings of IHM-HCI 2001*, Sept. 2001.



**Ivano Bartoli** received his Laurea degree in electronic engineering from the University of Rome "Roma Tre", Italy. He is currently a Research Engineer in Research and Innovation Department at Co.Ri.TeL, Rome, Italy. His research interests include image/speech compression and transport, video transcoding for real-time applications, traffic engineering for Ethernet/MPLS transport networks.



**Giovanni Iacovoni** received his Laurea degree in engineering from the University of Rome "La Sapienza" and Ph.D. from the University of Pavia, Italy. For many years he has worked in Ericsson Lab Italy, where he was the coordinator of the multimedia area in the Research and Innovation Department. Now he is with La Sapienza University, Rome, Italy.

He has worked in a number of EU projects and has coordinated the design and the development of different prototypes for multimedia performance evaluation. His current research interests include digital ASIC design, image/speech compression and transport, video transcoding for realtime applications, TCP analysis, traffic modeling.



**Fabio Ubaldi** received his Laurea degree in electronic engineering in 2002 from the University of Perugia, Italy.

He works in Co.Ri.TeL, where he is a technical researcher in the Research and Innovation Department. His research interests are in digital image/speech processing, audio/video transcoding for real-time applications, Linux embedded systems in real-time scenario.