Intermedia Synchronization Protocol for Continuous Media Using MPEG-4 in Mobile Distributed Systems

Eduardo Lopez Dominguez[1], Saul E. Pomares Hernandez[2], Pilar Gomez Gil[2], Jorge de la Calleja[3], Antonio Benitez[3] and Antonio Marin-Hernandez[4]
[1] Laboratorio Nacional de Informática Avanzada (LANIA)
Xalapa, Veracrux, 91000, Mexico
[2]National Institute of Astrophysics, Optics and Electronics (INAOE)
Tonantzintla, Puebla ; 72840, Mexico
[e-mail: spomares@inaoep.mx, pgomez@inaoep.mx]
[3]Universidad Politécnica de Puebla (UPP),
San Mateo, Puebla, 72640, Mexico
[e-mail: jdelacalleja@uppuebla.edu.mx, abenitez@uppuebla.edu.mx]
[4] Universidad Veracruzana
Xalapa, Veracrux, 91000, Mexico
[e-mail: elopez@lania.mx, anmarin@uv.mx]
*Corresponding author: Eduardo Lopez Dominguez

Received December 20, 2011; revised May 23, 2012; accepted June 13, 2012; published June 25, 2012

Abstract

The preservation of temporal dependencies among a group of processes that exchange continuous media at runtime is a key issue for emerging mobile distributed systems (MDS), such as monitoring of biosignals and interactive multiuser games. Although several works are oriented to satisfy temporal dependencies, most of them are not suitable for MDSs. In general, an MDS is characterized by the absence of global references (e.g. shared memory and wall clock), host mobility, limited processing and storage capabilities in mobile hosts, and limited bandwidth on wireless communication channels. This paper proposes an asymmetric synchronization protocol to be used at runtime in an MDS without using a common reference. One main aspect of our synchronization protocol is that it translates temporal constraints to causal dependencies of the continuous media data as seen by the mobile hosts. We simulate the protocol by considering a cellular network environment and by using MPEG-4 encoders. The simulation results show that our protocol is effective in reducing the synchronization error. In addition, the protocol is efficient in terms of processing and storage costs at the mobile devices, as well as in the overhead attached per message across the wired and wireless channels.

Keywords: Asymmetric, causal, mobile distributed system, synchronization mechanism

http://dx.doi.org/10.3837/tiis.2012.06.011

1. Introduction

The development of mobile distributed systems (MDS) that exchange continuous media at runtime among a group of processes has become a relevant research field. Examples of these systems include remote monitoring of biosignals for telemedicine networks and interactive multiuser games [1][2]. An important characteristic that the MDS must fulfill is the synchronization of continuous media data in multi-party environments. The problem of synchronization in this context concerns the preservation of temporal dependencies among the continuous data (e.g. audio and video) from the time of generation to the time of presentation at the group of processes.

Numerous works have proposed mechanisms to carry out the synchronization at runtime [2][3][4][5][6]. Nevertheless, these works cannot be applied in mobile distributed systems due to the fact that they do not consider the inherent characteristics of such systems, such as: absence of global references (e.g. shared memory and wall clock), host mobility, limited processing and storage capabilities in mobile hosts, and limited bandwidth on wireless communication channels.

In this work, we propose an asymmetric and distributed synchronization protocol to be used at runtime among a group of processes in an MDS. The synchronization protocol avoids the use of a common reference by using the logical mapping model (LMM) presented in [6]. Based on the LMM our protocol translates temporal constraints to causal dependencies of the continuous media data transmitted in an MDS. The protocol satisfies temporal constrains by ensuring data causal delivery dependencies when possible. One important aspect of our work is that the temporal constraints are satisfied according to the view of the end mobile hosts and not according to some intermediate element (e.g. proxy).

We did the simulation of the protocol considering a cellular network environment and by using MPEG-4 encoders. We show in this paper that our protocol is effective in reducing the synchronization error in an MDS without requiring previous knowledge of the system nor global references.

The rest of this paper is organized as follows. In section 2, we review the related work. Section 3 presents some preliminaries. In section 4, we describe the synchronization protocol. Section 5 presents the simulation results. Finally, some conclusions are presented in Section 6.

2. Related Work

Many works have proposed mechanisms to achieve the synchronization between streams [7][8][9][10][11][12]. We organize these works into three categories according to the design principle used: *end-to-end*, *proxy* and *asymmetric*. The end-to-end mechanisms were developed for distributed systems considering "smart" static hosts and wired communication channels [7][8].

The second category considers "dumb" communicant end hosts and for this reason, introduces a logical entity, called proxy, that carries out the main activities of the end hosts. The works in this category at an MDS reduce the overhead sent at wireless communication channels. However, these works introduce extra delays during the delivery of messages (known as unnecessary inhibition), and they cannot reflect the behavior seen of the mobile hosts because they accomplish the message delivery at and according to the view of the communication service points (e.g. base stations) [9][10].

The last category uses the asymmetric design principle [11][12]. Asymmetric protocols also discharge some tasks of the mobile hosts towards logical or physical entities, which are assumed to have more capacity. The main difference between this category and the previous one is that a direct logical channel is considered to exist between the end hosts [13][14]. This characteristic is important since it allows a communication and interaction according to the view of the mobile hosts. Our work is classified in this category. To the best of our knowledge, our work is the first asymmetric protocol to synchronize continuous media with causality control at mobile distributed systems.

3. Preliminaries

3.1 The System Model

Mobile host (MH) and Base station (BS): The MDS under consideration runs over a wireless network with infrastructure, which is composed by a set of mobile hosts $P=\{p_1, p_2, \ldots p_n\}$ and a set base stations $BSG= \{BS_1, BS_2, \ldots BS_s\}$.

Messages: We consider a finite set of messages M, where each message $m \in M$ is identified by a tuple $m=(p, x)$, where $p \in P$ is the sender mobile host of m, and x is the local logical clock for messages of p, when m is broadcasted. The set of destinations of a message m is always P.

Events: Let m be a message. We denote by $send(m)$ the emission event and by $delivery(p,m)$ the delivery event of m to mobile host $p \in P$. The set of events associated to M is the set $E = \{send(m) : m \in M\} \cup \{delivery(p,m) : m \in M \wedge p \in P\}$. The mobile host $p(e)$ of an event $e \in E$ is defined by $p\ (send(m)) = p$ and $p\ (delivery(p,m)) = p$. The set of events of a mobile host p is $E_p = \{e \in E : p(e) = p\}$.

Intervals: We consider a finite set I of intervals, where each interval $A \in I$ is a set of messages $A \subseteq M$ sent by mobile host $p = Part(A)$, defined by the mapping $Part : I \rightarrow P$. We denote by a^- and a^+ the endpoint messages of A, and due to the sequential order of $Part(A)$, we have that for all $m \in A : a^- \neq m$ and $a^+ \neq m$ implies that $a^- \rightarrow m \rightarrow a^+$.

3.2 Background and Definitions

The Happened-Before Relation (HBR). Causal ordering delivery is based on the causal precedence relation defined by [15]. The partial order is defined as follows:

Definition 1. The causal relation "$\rightarrow$" is the least partial order relation on E satisfying the following properties:

1) If a and b are events belonging to the same process, and a was originated before b, then $a \rightarrow b$.
2) If a is the sent message of a process, and b is the reception of the same message in another process, then $a \rightarrow b$.
3) If $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$.

By using Definition 1, a pair of events is said to be concurrently related "$a \parallel b$" if and only if $\neg (a \rightarrow b \vee b \rightarrow a)$.

The Immediate Dependency Relation (IDR) [16] is the transitive reduction of the HBR. We denote it by $\downarrow$, and it is defined as follows:

Definition 2. Immediate Dependency Relation "$\downarrow$" (IDR):

$$e \downarrow e' \Leftrightarrow [\ (e \rightarrow e') \wedge \forall\ e'' \in E, \neg(e \rightarrow e'' \rightarrow e')]$$

Thus, an event *e* directly precedes an event *e'*, if and only if no other event *e''* belonging to *E* exists, such that *e''* belongs at the same time to the causal future of *e* and to the causal past of *e'*.

Broadcast Causal Delivery. The causal delivery for group communication (broadcast case) based on the IDR is defined as follows:

$$\text{If } \forall\ m,m' \in M, send(m) \downarrow send(m') \Rightarrow \forall\ p \in P\text{: } delivery(p,m) \rightarrow deliver(p,m') \text{ then}$$
$$\forall\ m,m' \in M, send(m) \rightarrow send(m') \Rightarrow \forall p \in P\text{: } delivery(p,,m) \rightarrow delivery(p,m')$$

The Partial Causal Relation (PCR) was introduced by Fanchon et al. [17]. It considers a subset $M' \subseteq M$ of messages. The PCR induced by *M'* takes into account the subset of events $E' \subseteq E$ that refer to *send* or *delivery* events of the messages belonging to *M'*. Formally, the PCR was defined as follows:

Definition 3: The partial causal relation '$\rightarrow_{E'}$' is the least partial order relation satisfying the two following properties:

1) For each participant $p \in P$, the local restrictions of $\rightarrow_{E'}$ and $\rightarrow$ to the events of E'_p coincide: $\forall e, e' \in E'_p : e \rightarrow e' \Leftrightarrow e \rightarrow_{E'} e'$
2) For each message $m \in M'$ and $j \in P$, the emission of *m* precedes its delivery to $j : j \in P \Rightarrow send(m) \rightarrow_{E'} delivery(j,m)$.

The Happened-Before Relation on Intervals. The two relations identified for intervals are the causal relation and the simultaneous relation. For further details concerning these definitions, refer to [6]. We begin by giving the definition of the causal relation applied to intervals. Formally, it is defined as:

Definition 4. The relation " $\rightarrow_I$ " is accomplished if the following two conditions are satisfied:

1) $A \rightarrow_I B$ if $a^+ \rightarrow_{E'} b^-$
2) $A \rightarrow_I B$ if $\exists C \mid (a^+ \rightarrow_{E'} c^- \wedge c^+ \rightarrow_{E'} b^-)$

where a^+ and b^- are the *end* and *begin* events (or messages) of *A* and *B* respectively, c^- and c^+ are the endpoints of *C*, and $\rightarrow_{E'}$ is the partial causal order induced on $E' \subseteq E$, where *E*', in our case, is the subset composed by the endpoint messages of the intervals in *I*.

Definition 5. Two intervals *A*, *B* are said to be simultaneous "$|||$" if the following condition is satisfied:

$$A \mid\mid\mid B \Rightarrow a^- \mid\mid b^- \wedge a^+ \mid\mid b^+$$

The definition above means that one interval *A* can take place at the 'same time' as another interval B. Finally, the definition of causal delivery for intervals based on their endpoints is presented as follows:

Definition 6. Causal Broadcast Delivery for Intervals based on the endpoints:

$$\text{If } (a^+, b^-) \in A\times B,\ send(a^+)\rightarrow_{E'} send(b^-) \Rightarrow \forall\ p\in P,\ delivery(p,\ a^+) \rightarrow_{E'} delivery(p,b^-) \text{ then } \forall p\in P\ delivery(p,A) \rightarrow_I delivery\ (p,B)$$

3.3 Temporal Synchronization Model

The model used to represent the interaction among continuous media data is the logical mapping model introduced in [6]. The logical mapping model specifies the synchronization of a temporal scenario based on the identification of logical precedence among the media involved. Specifically, a logical mapping translates an interval temporal relation to be expressed in terms of the happened-before relation on intervals.

Table 1. Logical mappings expressed on endpoints

Logical Mappings	**Expressed on endpoints**
precedes: $A\rightarrow_I B$	$a^+\rightarrow b^-$
simultaneous: $C \mid\mid\mid D$	$c^- \mid\mid d^-,\ c^+ \mid\mid d^+$
ends: $A\rightarrow_I(C \mid\mid\mid D)$	$a^+\rightarrow c^-,\ a^+\rightarrow d^-,$ $c^- \mid\mid d^-,\ c^+ \mid\mid d^+,$
starts: $(C \mid\mid\mid D)\rightarrow_I B$	$c^- \mid\mid d^-,\ c^+ \mid\mid d^+,$ $c^+\rightarrow b^-,\ d^+\rightarrow b^-$
overlaps: $A\rightarrow_I(C \mid\mid\mid D)\rightarrow_I B$	$a^+\rightarrow c^-,\ a^+\rightarrow d^-,$ $c^- \mid\mid d^-,\ c^+ \mid\mid d^+,$ $c^+\rightarrow b^-,\ d^+\rightarrow b^-$

The logical mapping model identifies five logical mappings which are: *precedes*, *simultaneous*, *ends*, *starts* and *overlaps* (see **Table 1**). These five logical mappings are sufficient to represent all possible continuous media temporal relations (interval-interval relations). The resulting logical mappings represent synchronization specification units, which are automatic processable descriptions of a given temporal relation.

4. Synchronization Protocol

Our synchronization protocol carries out two main tasks at runtime: first, it performs the translation of continuous media according to the temporal model previously presented in order to establish their interval temporal relations; secondly, it carries out the presentation of the intervals based on the resultant logical mapping. Internally, the synchronization protocol uses causal and FIFO messages. The causal messages are divided into: *begin*, *cut* and *end* messages. The *begin* and *end* messages are the left and right endpoints of the original intervals, and cut is a control message used to notify about an interval segmentation. The FIFO messages (called *fifo*) are only used inside an interval. We note that a causal message also locally satisfies the FIFO order. Next, we describe the main structures and components of the protocol.

4.1 Data Structures

Each **mobile host** p uses and stores the following data structures:

- *mes_received*(p) is a counter, which is incremented by p each time a causal or FIFO message is received.
- *mes_sent*(p) is a counter, which is incremented each time a causal or FIFO message is sent by the mobile host p.
- *CausalMesDM*(p) is a counter, which is incremented each time a causal message is received by the mobile host p.
- *FIFOMesDM*(p) is a counter which is incremented each time a FIFO message is received by mobile host p.
- $\Phi(p)$ is a structure of bits. Each *bit* in $\Phi(p)$ identifies a message m in the causal past of p. The size of $\Phi(p)$ fluctuates between $1 \leq |\Phi(p)| \leq n\text{-}1$. This structure is the only control information attached to each causal message sent in the wireless channels.

Each **base station *BS*** uses and stores the following data structures:

- *mes_sent*(*BS*) is a counter, which is incremented each time a causal or FIFO message is sent by the base station *BS* in its cell.
- *causalMes_sent*(*BS*) is a counter, which is incremented each time a causal message is sent by the base station *BS*.
- *fifoMes_sent*(*BS*) is a counter which is incremented each time a FIFO message is sent by the base station *BS*.
- *VT*(*BS*) is the vector time. For each mobile host p, there is an element *VT*(*BS*)[i], where i is the mobile host identifier of p_i.
- *CI*(*BS*) is the control information structure. It is a set of entries (i, t, d). The entry (i, t, d) represents a message sent by the mobile host p_i with logical local clock t=*VT*(*BS*)[i] and d=*causalMes_sent*(*BS*).
- *last_fifo*(*BS*) has information about the last *fifo* messages received by *BS*. It is a set of entries (i, t, f) where f=*fifoMes_sent*(*BS*). This structure is important because it represents potential causal messages.

Message structures. The following message structures are used by the mobile hosts and base stations.

- A message sent in the wireless communication channels by mobile hosts to their *BS* is identified by m, and has the following structure: m≡(i, t, *mes_received*(p), *TP*, *data*, $h(m)$), where:
  – *TP* is the type of message (*begin*, *end*, *cut* and *fifo*).
  – $h(m)$ is a structure of bits created at the moment of transmission of a message m by p_i, which is a copy of $\Phi(p)$.
- A message m sent among base stations *BS*s is denoted by $bs(m)$, and it is composed by a quintuplet $bs(m) \equiv$ (i, t, *TP*, *data*, $H(m)$), where:
  – *data* is the content of the message, and
  – $H(m)$ is composed of a set of elements (i, t), which represent messages in the causal past of p_i. A structure $H(m)$ is created when a broadcast message is sent by a base station.

- A message *m,* received by a BS_l from a mobile host $p \in BS_l$ and which has been resent by such BS_l in its cell, consists of a quintuplet that we call *intra*(*m*)≡ (*i, t*, *TP, data, h'*(*m*)), where:
  - *h'*(*m*) is a structure of bits, which is created at the moment of transmission of a message *m* by base station BS_l.
- A message *bs*(*m*) received by a BS_l and which has been resent within its cell consists of a quintuplet that we call *inter*(*m*) ≡ (*i, t*, *TP, data, h'*(*m*)).

4.2 Specification of the Synchronization Protocol

Table 2. Initialization

Structures and variables at mobile host p_i, i: 1...n;	**Structures and variables at base station BS_l, l:1...s;**
$\Phi(p_i) \leftarrow \emptyset$	$VT(BS_l)[i]=0 \forall i:1\ldots n;$
$bit \leftarrow 1\text{b}$	$CI(BS_l) \leftarrow \emptyset$
$mes\_received(p_i) = 0$	$last\_fifo(BS_l) \leftarrow \emptyset$
$mes\_sent(p_i) = 0$	$H(m) \leftarrow \emptyset$
$CausalMesDM(p_i)=0$	$mes\_sent(BS_l) = 0$
$FIFOMesDM(p_i)=0$	$causalMes\_sent(BS_l) = 0$
	$fifoMes\_sent(BS_l) = 0$

Table 3. Sending of messages *begin*, *end*, *cut* and *fifo* by the mobile host p_i

1.	Send(Input: $TP = \{ begin \mid end \mid cut \mid fifo \}$)
2.	$mes\_sent(p_i) = mes\_sent(p_i) + 1$
3.	$h(m) \leftarrow \Phi(p_i)$
4.	**if** not ($TP == fifo$) then
5.	**if** not ($TP == begin$) then
6.	$m=(i=id\_mobile, t= mes\_sent(p_i), TP, CausalMesDM(p_i), data, h(m))$
7.	**if** ($TP == end$) then
8.	$Act = false$ **endif**
9.	**else**
10.	$m=(i=id\_mobile,t=mes\_sent(p_i),TP,CausalMesDM(p_i), FIFOMesDM(p_i),data\ h(m))$
11.	$Act = true$ **endif**
12.	$\Phi(p) \leftarrow \emptyset$
13.	$CausalMesDM(p_i) = CausalMesDM(p_i) + 1$
14.	**else**
15.	$m=(i=id\_mobile,t=mes\_sent(p_i),TP,data,h(m) \leftarrow \emptyset)$ **endif**
16.	**Diffusion :** sending(m)

Table 4. Reception of message *intra*(*m*)|*inter*(*m*)=(*i*, *t*, *TP*, *data*, *h'*(*m*)) at the mobile host p_j, $i \neq j$

1.	**if** not ($t == mes\_received(p_j) + 1$) then
2.	*wait* (*intra*(*m*)| *inter*(*m*))
3.	**else**
4.	*delivery*(*intra*(*m*)| *inter*(*m*))
5.	$mes\_received(p_j) = mes\_received(p_j) + 1$
6.	**if** not ($TP == fifo$) then

7.	$CausalMesDM(p_j) = CausalMesDM(p_j) + 1$
8.	$\forall \{ bit\} \in h'(m)$
9.	$\Phi(p_j) \leftarrow \Phi(p_j) / \{bit\}$
10.	$\Phi(p_j) \leftarrow \Phi(p_j) \cup \{ bit\}$
11.	**if**($act{=}{=}true$) and not($TP == cut$) and not($TP == begin$) then
12.	$send(cut)$ **endif**
13.	**else**
14.	$FIFOMesDM(p_j) = FIFOMesDM(p_j) + 1$ **endif**
15.	**endif**

Table 5. Reception-sending of a message m=(i, t, TP, $CausalMesDM(p_i)$, $data$, $h(m)$) at base station BS_l

1.	**if** $i \in BS_l$ then
2.	**if** not ($t == VT(BS_l)[i] + 1$) then
3.	$wait(m)$
4.	**Else**
5.	$delivery(m)$
6.	$VT(BS_l)[i] = VT(BS_l)[i] + 1$
7.	**if** not ($TP == fifo$) then
8.	**if** not ($TP == begin$) then
9.	$\forall \{bit\} \in h(m)$
10.	**if** $\exists(k,t,d) \in CI(BS_l) \mid d == CausalMesDM(p)$ and $ip == false$ then
11.	$h'(m) \leftarrow h'(m) \cup \{bit\}$
12.	$(k, t, d) \leftarrow (k, t, d)$ **endif**
13.	**if** $\exists(k,t,d) \in CI(BS_l) \mid d == CausalMesDM(p)$ then
14.	$H(m) \leftarrow H(m) \cup (k, t)$ **endif**
15.	$CausalMesDM(p) = CausalMesDM(p) - 1$
16.	**else**
17.	**for** each $\{bit\} \in h(m)$ do
18.	**if** $\exists(s,r,d) \in last\_fifo(BS_l) \mid d \leq FIFOMesDM(i)$ and $s \mathrel{!=} j$ then
19.	$h'(m) \leftarrow h'(m) \cup \{bit\}$
20.	$H(m) \leftarrow H(m) \cup (s, r)$
21.	$j = s$ **endif, endfor**
22.	$\forall (k,t,d) \in CI(BS_l)$
23.	**if** $\exists(s,r) \in H(m) \mid k{=}s$ and $d \leq CausalMesDM(p)$ then
24.	**if** not($(s,r) == max((k,t),(s,r))$) then
25.	$H(m) \leftarrow H(m) \cup (k,t)$ **endif**
26.	$(k, t, d) \leftarrow (k, t, d)$ **endif**
27.	**endif**
28.	$causalMes\_sent(BS_l) = causalMes\_sent(BS_l) + 1$
29.	$CI(BS_l) \leftarrow CI(BS_l) \cup \{(i, t, causalMes\_sent(BS_l))\}$
30.	**endif**
31.	**endif**
32.	**if** ($TP == fifo$) then

```
33.       $H(m) \leftarrow h'(m) \leftarrow \emptyset$
34.       $fifoMes\_sent(BS_l) = fifoMes\_sent(BS_l) + 1$
35.       $last\_fifo(BS_l) \leftarrow last\_fifo(BS_l) \cup \{i,t,fifoMes\_sent(BS_l\}$ endif
36.     $mes\_sent(BS_l) = mes\_sent(BS_l) + 1$
37.     $intra(m) = ( i, t= mes\_sent(BS_l), TP, data, h'(m) )$
38.     $bs(m) = ( i, t, TP, data, H(m))$
39.     Diffusion : $send(intra(m))$
40.     Diffusion : $send(bs(m))$  endif
```

Table 6. Reception-sending of message $bs(m)$=(i, t, TP, $data$, $H(m)$) by the base station BS_l

```
1.  if $i \notin BS_l$ then  /*A $BS$ can receive its own message $bs(m)$*/
2.     if not (($t == VT(BS_l)[i] + 1$)  and  ($\forall (s,x) \in H(m) : x \leq VT(BS_l)[s]$ ) ) then
3.          $wait(bs(m))$
4.       Else
5.          $delivery(bs(m))$
6.          $VT(BS_l)[i] = VT(BS_l)[i] + 1$
7.          if not ( $TP == fifo$) then
8.             $causalMes\_sent(BS_l) = causalMes\_sent(BS_l) + 1$
9.             $\forall (x,y) \in H(m)$  /*Construction of the structure of bits */
10.            if $\exists$ $(k,t,d) \in CI(BS_l)$ | $last\_fifo(BS_l)$ | $x == k$ and $y == t$
11.                 $h'(m) \leftarrow h'(m) \cup \{bit\}$
12.                 $(k,t,d) \leftarrow (k,t,d)$     endif
13.            $CI(BS_l) \leftarrow CI(BS_l) \cup \{i,t,causalMes\_sent(BS_l)\}$
14.         endif
15.    endif
16. endif
17.  if ( $TP = fifo$) then
18.     $H(m) \leftarrow h'(m) \leftarrow \emptyset$
19.     $fifoMes\_sent(BS_l) = fifoMes\_sent(BS_l) + 1$
20.     $last\_fifo(BS_l) \leftarrow last\_fifo(BS_l) \cup \{i,t,fifoMes\_sent(BS_l)\}$ endif
21.  $mes\_sent(BS_l) = mes\_sent(BS_l) + 1$
22.  $inter(m) = ( i, t= mes\_sent(BS_l), TP, data, h'(m) )$
23.  Diffusion : $send(inter(m))$
```

4.3 Description

In order to explain how our synchronization protocol carries out the process of logical mapping in a wireless network with infrastructure, the scenario of **Fig. 1** is presented. Recall that this process in our work is made online by identifying the causal boundaries of the concerned segment(s) from left to right. In the example, segment A must first be determined. To achieved this, we need to map the left causal boundary a^- as equal to $x^-=x_1$, and the right causal boundary as equal to $a^+=x_k$. This is carried out by our mechanism as follows. When the message $inter(a^-)$= (p_1, x_1,$begin$,$data$, $h'(m$)=$\emptyset$) is received at the mobile host p_2, it will be delivered as soon as the FIFO delivery condition is satisfied (Line 1, **Table 4**). This condition ensures that $inter(a^-)$ will be delivered if and only if all its causally related messages have been delivered. Later, the mobile host p_2 determines that x_1 is the $begin$ causal message a^- (Lines 6, **Table 4**), and it updates $\Phi(p_2)$ with the attached information of $inter(x_1)$ (Lines 8-10, **Table 4**). The resulting structure of bits is $\Phi(p_2)$=1b.

Then, the right endpoint a^+ is determined by the last message received by mobile host p_2 before *begin* send event *send*(d^-). In order to determine a^+, at the diffusion of message *begin* d^- by p_2 (Lines 10-12, **Table 3**), the mobile host p_2 copies the structure $\Phi(p_2)$ to $h(d^-)$ (Lines 3, **Table 3**). Afterward, the message d^- = (p_2, y_1, *begin*, *CausalMesDM*(p_2), *FIFOMesDM*(p_2), *data*, $h(d^-)$=1b) is constructed and sent in Lines 10-12 and 16, **Table 3**. Through $h(d^-)$, *FIFOMesDM*(p_2) and *last_fifo*(BS_2), the local base station BS_2 will be able to determine that the last received message by p_2 before the *begin* send event *send*(d^-) is equal to x_k, Lines 16-27, **Table 5**. Once the causal boundaries of A are known, we can determine the set of messages that compose it (A=$\{x_1, x_2,\ldots,x_k\}$).

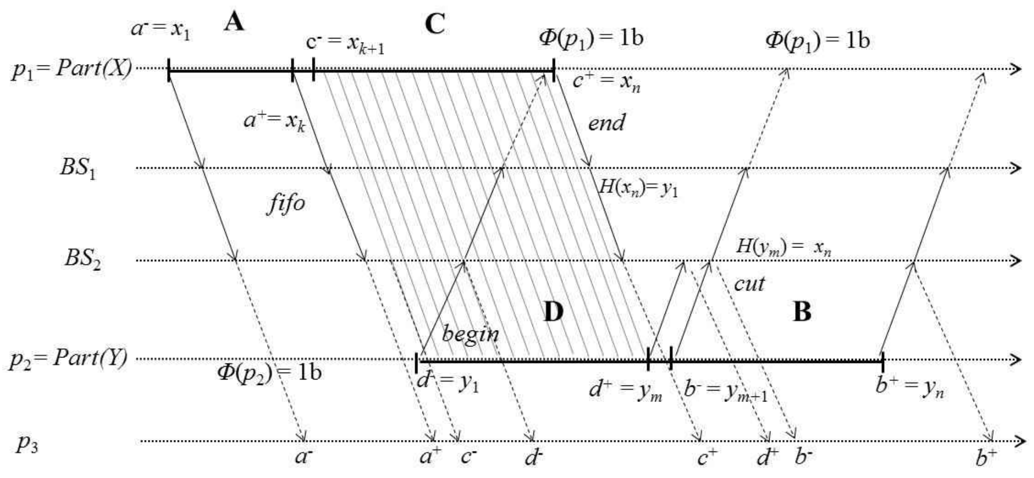


Fig. 1. Construction of logical mapping $A \rightarrow_I (C|||D) \rightarrow_I B$ of the overlaps relation.

After interval A is identified, we proceed to identify the causal boundaries of intervals C and D. At this point, we can map the left causal boundaries c^- and d^- with the *send* events of x_{k+1} and y_1, respectively. But, it is only until the *send* and *delivery* events of the right endpoint x^+ take place, that we can identify the right endpoints of C and D. With the *end* send event *send*($x^+ = x_n$), we establish that $c^+ = x_n$ (Lines 5-8, **Table 3**), and consequently, $C = \{x_{k+1}, x_{k+2}, \ldots, x_n\}$. When message *inter*($x_n$) is received by the mobile host p_2, it determines that x_n is the *end* causal message c^+ in Line 6, **Table 4**. As a direct result, the mobile host p_2 sends a *cut* message which establishes the end of interval D($d^+ = y_m$) and the beginning of interval B (*cut* = $b^- = y_{m+1}$), Lines 11-12, **Table 4**. Hence, we have D=$\{y_1, y_2, \ldots, y_m\}$. The diffusion of *cut* message by p_2 is carried out as follows. First, the mobile host p_2 copies the structure $\Phi(p_2)$ to $h(y_{m+1})$ (Lines 3, **Table 3**) and it constructs and sends message y_{m+1}=(p_2,1, *cut*, *CausalMesDM*(p_2), *FIFOMesDM*(p_2), *data*, $h(y_{m+1})$=1b) in Lines 10-12 and 16, **Table 3**. In this case, through $h(y_{m+1})$ and *CausalMesDM*(p_2) the local base station BS_2 will be able to determine which messages immediately precede y_{m+1}, Lines 8-16, **Table 5**. Therefore, the resulting message is $bs(b^-)$≡(p_2, y_{m+1},*cut*,*data*, $H(b^-)$=(p_1, x_n)).

Afterward, when the message $bs(b^-)$ is received by the base station BS_1, its FIFO and causal delivery conditions are verified. In this case, $bs(b^-)$ satisfies both conditions (Line 2, **Table 6**). Therefore, $bs(b^-)$ is delivered and the vector is increased by one in $VT(BS_1)[p_2]$. Later on, BS_1 sends the message $bs(b^-)$ to its local mobile hosts. The message sent is *inter*(b^-)=(p_2,y_{m+1},*data*, $h'(b^-)$=1b), (Line 22, **Table 6**). Finally, with the send event of y^+, we have $b^+ = y_m$, and consequently, $B = \{y_{m+1}, y_{m+2}, \ldots, y_n\}$.

Therefore, the resultant logical mapping for the scenario presented in **Fig. 1** is $A \rightarrow_I (C||D) \rightarrow_I B$. To carry out the interval causal delivery at the mobile host p_3 in terms of their endpoints, we only need to ensure that:

- $delivery(p_3, a^+) \rightarrow_{E'} delivery(p_3, c^-)$,
- $delivery(p_3, a^+) \rightarrow_{E'} delivery(p_3, d^-)$,
- $delivery(p_3, c^+) \rightarrow_{E'} delivery(p_3, b^-)$ and
- $delivery(p_3, d^+) \rightarrow_{E'} delivery(p_3, b^-)$

4.4 Correction of Synchronization Error

Each time that a base station receives a causal message, it carries out two corrective actions. First, a *BS* delays the delivery of messages when they do not satisfy the causal dependencies; and secondly, it discards messages when the maximum waiting time is exceeded. In order to determine how much time a causal message *m* must wait from its reception to its delivery, we define the following function:

$wait(bs(m))$: $\forall\ m' = (k,s') \in H(m), \{$
a) $s' \leq VT(BS_l)[k]$ or
b) $deadline_l\ (m') \leq current\_time(BS_l)\ \}$

where $deadline_l(m')$ returns the expiration time of message m' at BS_l and $current\_time(BS_l)$ has the present time at the same base station. We note that if the second condition is satisfied, then message m' is discarded by BS_l, and every message $m'' = (k, s'')$ such that $s'' > VT(BS_l)[k]$ and $s'' < s'$ is also discarded. The function $wait(bs(m))$ can be inserted at line 3 of **Table 6**. The deadline of a message m', denoted by $deadline_l(m')$, can be determined through a centralized method [18] or a distributed method. In our work, we use the distributed method proposed by Pomares et al. in [19]. Here, we do not present how the deadline of a message is calculated; you can refer to this work for details.

4.5 Overhead Analysis

In our work, the size of the control information sent over a wired channel depends on the number of concurrent messages that immediately precede a causal message *m*. Since $H(m)$ has only the most recent causal messages that precede a message *m*, the overhead per causal message to ensure synchronization is given by the cardinality of $H(m)$, which can fluctuate between 0 and n-1 ($0 \leq |\ H(m)| \leq n-1$), where *n* is equal to the number of mobile hosts in the group. In the best case, dealing with the serial case (no concurrency of messages exists), we note that the message overhead is equal to 1, and in the case of concurrent messages, the worst case is n-1. On the other hand, in our protocol, the control information attached to causal messages sent over a wireless channel and stored at a mobile host is $1 \leq |h(m)| \leq n-1$, where $h(m)$ is a structure of bits. Again, the size of $h(m)$ depends on the number of concurrent messages that immediately precede a causal message. In the best case, the size of $h(m)$ is equal to 1 bit and in the worst case, the size of $h(m)$ is equal to n-1 *bits*, where *n* is equal to the number of mobile hosts in the group.

5. Simulation

In order to evaluate the synchronization protocol developed in this work, we carry out the simulation of our mechanism at a mobile distributed system. In our work, the considered scenario simulates a mobile distributed system running over a cellular network. For this, we use the simulator Sinalgo [20]. The scenario considered is integrated by four base stations and four mobile hosts. The geographic area covered by a base station (BS) is called a *cell*. At any given time, a mobile host (MH) is assumed to be within the cell of at most one base station. An MH can communicate with other mobile hosts only through its BS. A BS communicates with mobile hosts through wireless channels. The BSs are connected among themselves by using wired channels. We assume that the wired and wireless channels are reliable and asynchronous, see **Fig 2**.

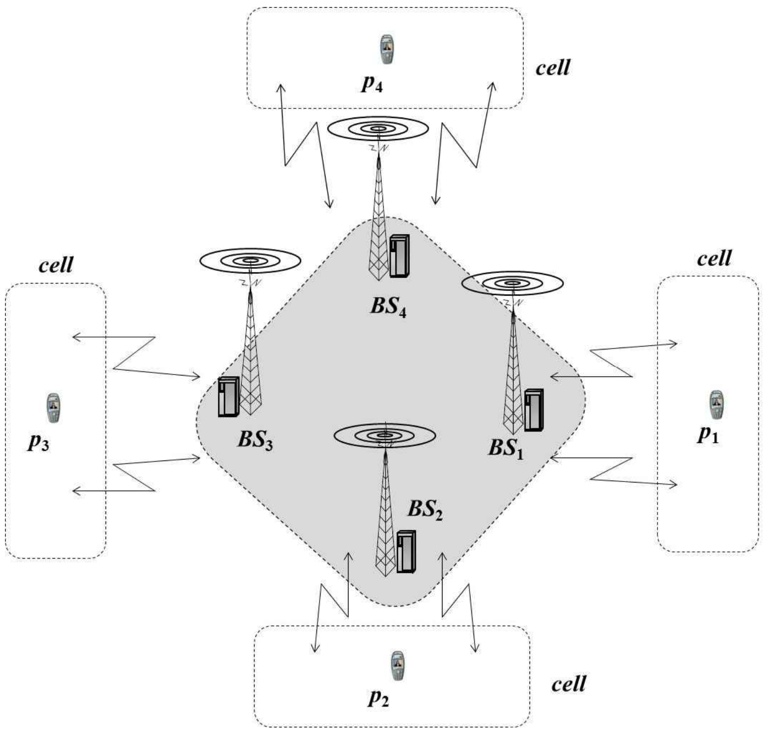


Fig. 2. Simulation architecture

The mobile hosts only transmit one media (audio or video). For the audio and video, we use MPEG-4 encoders of the MPEG4IP project [21]. For the case of audio, the silence compression MPEG-4 CELP mode is used. In this mode, we send a *begin* message each time that the TX_Flag equals to 1 (indicates voice activity), and a frame is send as an *end* message each time that TX_Flag ≠ 1 (indicates no or low voice activity). The remaining audio frames are sent as *fifo* messages. In the case of video, it was encoded as a single video object into a single video object layer at a rate of 25 frames/s, the Group of Pictures (GoP) pattern is set to IBBPBBPBBPBB. The I frames are sent as *begin* messages, and the last B frames of the patterns are sent as *end* messages. The rest of the video frames are sent as *fifo* messages.
In our scenario, we assume that the mobile hosts do not share a common physical clock or a

common memory. Using Sinalgo, we establish different configurations for the communication channels, which reflect different network conditions; specifically, we use different configurations for transmission delay, see section 5.2 .

5.1 Measuring the synchronization error

In applications that use continuous data streams in real time among a group of participants, one of the main aspects to evaluate is the inter-stream synchronization error experienced by the mobile hosts. For example, in a videoconference the highest inter-stream synchronization error allowed between the sources is about 400ms [3]. In order to measure the inter-stream synchronization error, we define the following equations: *rcv_syn_error*$_l$(*m*) and *dlv_syn_error*$_l$(*m*). The equation *rcv_syn_error*$_l$(*m*) calculates the synchronization error experienced by BS_l at the reception of a message *bs*(*m*). On the other hand, the equation *dlv_syn_error*$_l$(*m*) measures the synchronization error experienced by BS_l at the delivery after applying our synchronization protocol. The definition of the equations is presented below. The equation *rcv_syn_error*$_l$(*m*) is defined as follows :

$$rcv\_syn\_error_l(m) = \frac{\sum_{(k,s)\in H(m)} receive\_time_l(bs(m)) - last\_rcv_l(k)}{|H(m)|} \tag{1}$$

The equation *dlv_syn_error*$_l$(*m*) is defined as follows:

$$dlv\_syn\_error_l(m) = \frac{\sum_{(k,s)\in H(m)} delivery\_time_l(bs(m)) - last\_dlv_l(k)}{|H(m)|} \tag{2}$$

where *last_rcv*$_l$(*k*) and *last_dlv*$_l$(*k*) returns the time when the base station BS_l has received or delivered in its cell the last message sent by the mobile host *k,* respectively. The functions *receive_time*$_l$(*bs*(*m*)) and *delivery_time*$_l$(*bs*(*m*)) return the reception or delivery time of message *m* at the base station BS_l, respectively.

5.2 Simulation Results

In order to show the effectiveness and efficiency of our protocol, two experiments have been carried out. Each experiment is composed by one hundred tests. The first experiment considers that the transmission delay is between 50-150 ms, which reflects a non-congested cellular network. In the second experiment, the transmission delay considered is between 50-400 ms, which reflects a heavy network load. For simplicity, we consider that the transmission delay at wired and wireless communication channels is the same. In the two experiments, the synchronization error at the message reception, the synchronization error at message delivery and the message and storage overhead were evaluated. The results obtained are presented below.

Experiment 1. For the first set of tests, the synchronization error at message reception measured by Equation 1 surpassed 400 ms at various points (see the dotted line in **Fig. 3**). On the other hand, after applying our protocol, the synchronization error measured by Equation 2 at message delivery is reduced at most of the points to less than 80 ms (refer to the continuous line in **Fig. 3**), which is below of the tolerated synchronization error by a videoconference.

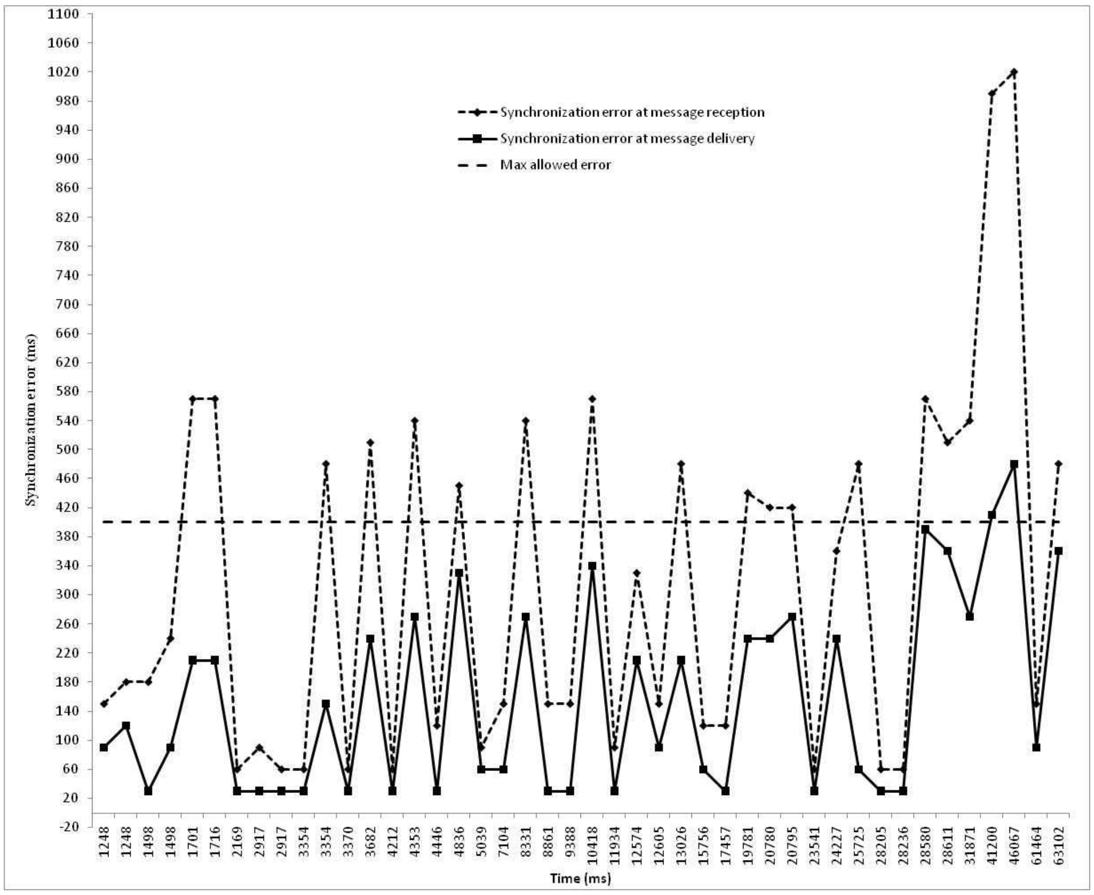


Fig. 3. Inter-stream synchronization error at reception and delivery of streams sent considering a random delay of 50-150ms.

On the other hand, in order to be efficient, the proposed protocol uses the specification of logical mappings based on *endpoints*. In our work, these endpoints are sent as causal messages (*begin*, *end* and *cut*). Therefore, the control information added (overhead) only corresponds to the causal messages sent (see Section 4.3). In this experiment the average overhead attached per causal message sent over a wired channel was around 8.2 bytes (see **Table 7**); and the average overhead per causal message sent over a wireless channel was only 2 bits.

Table 7. Message and storage overhead

Average per test	Number of causal and fifo messages		Average overhead per causal message		Average storage overhead	Total overhead of causal messages	
Experiment	Causal messages	fifo messages	Wired channel (bytes)	Wireless channel (bits)	Mobile host (bytes)	Wired channel (bytes)	Wireless channel (bytes)
1	3466	18437	7.9	2	8.3	27381	866.5
2	2737	17004	8.2	2	8.3	22444	684.3

In our protocol, a mobile host only uses and stores four counters and a bit structure (see Section 4.1). The size of the structure of bits depends on the number of concurrent messages that immediately precede a causal message. Therefore, the average storage overhead per test at the mobile hosts was around 8.3 bytes.

Experiment 2. For the second set of tests, we can see in **Fig. 4** that the synchronization error at message reception measured by Equation 1 exceeds in several points the maximum allowed error (see **Fig. 4**). This effect can produce, at an application level, the loss of coherency among the media played. In our work, after applying our asymmetric synchronization protocol, the inter-stream synchronization error, measured by Equation 2, perceived by the mobile hosts is less than 400ms in nearly all cases.

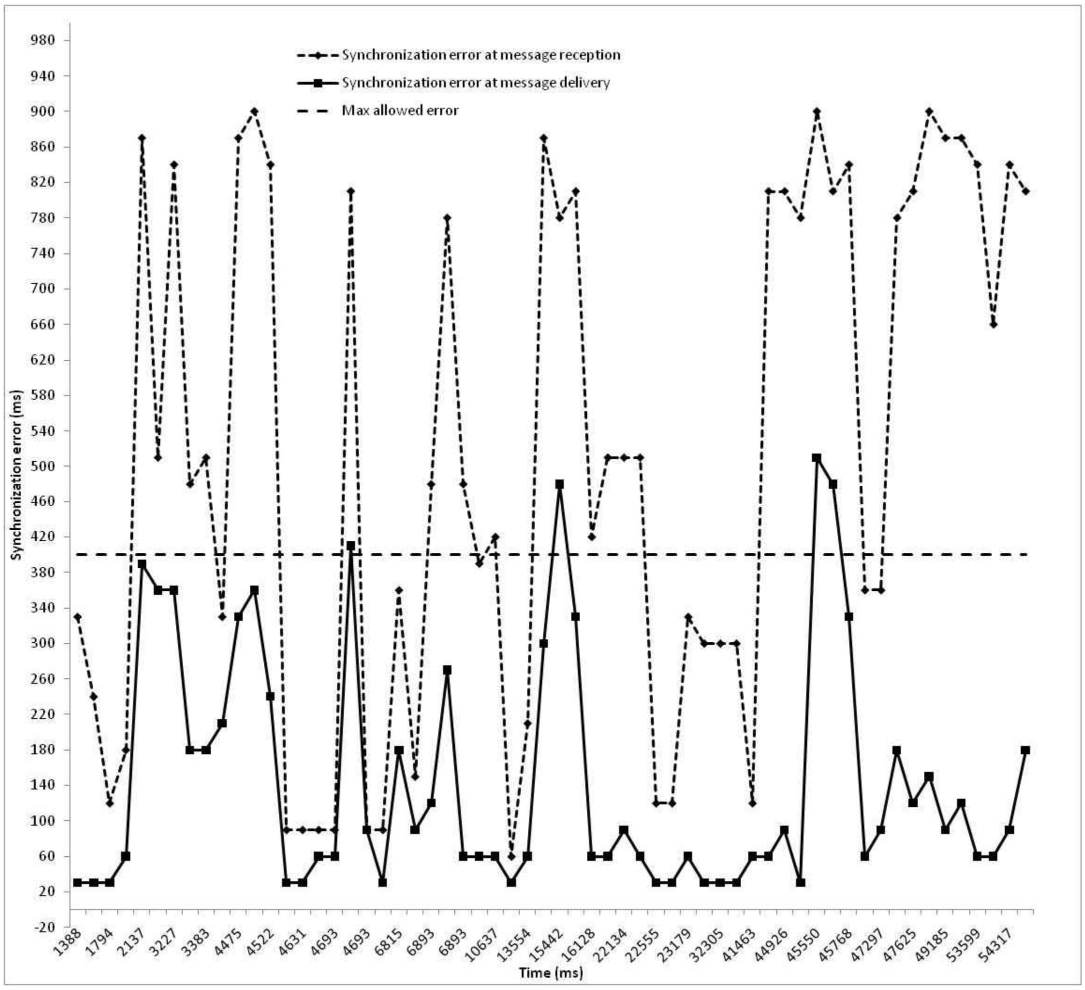


Fig. 4. Inter-stream synchronization error at reception and delivery of streams sent considering a random delay of 50-400ms.

On the other hand, in this experiment the average overhead attached per causal message sent over a wired channel was around 7.9 bytes. The average storage overhead at the mobile hosts was around 8.3 bytes and the average overhead per causal message sent over a wireless channel was only of 2 bits, see **Table 7**.

We note that for both experiments, the storage overhead at the mobile hosts and the average overhead per causal message sent over wireless channels remains small, see **Table 7**. Moreover, we note that the processing cost is also low at the mobile hosts since the updating process of the storage control information is based on bitwise operations and basic arithmetic operations (see **Table 4**).

6. Conclusions

We have presented a distributed synchronization protocol for continuous media to be used at runtime among a group of processes in an MDS. The protocol avoids the use of a global reference (e.g. wall clock) by executing temporal dependencies according to the causal dependencies of the continuous media data as viewed by the end mobile hosts. To demonstrate the effectiveness and efficiency of the synchronization protocol, we carried out the simulation of the protocol considering a group of processes, a cellular network architecture and MPEG-4 encoders. The simulation results show that our protocol reduces in all cases the synchronization error at the message delivery. Such correction in most of the cases gives as a result a tolerable error according to the maximum synchronization error supported. Furthermore, the simulation results show that the protocol is efficient in terms of processing and storage cost at the mobile devices, as well as is in the overhead attached per message across the wired and wireless channels.

References

[1] B. Fong, A. C. M. Fong, and C. K. Li, "Telemedicine Technologies: Information Technologies in Medicine and Telehealth", *1st Edition*, Wesley, USA, 2011. Article (CrossRef Link)

[2] E. Cronin, B. Filstrup, S. Jamin, and A.R. Kurc, "An efficient synchronization mechanism for mirrored game architectures", *Multimedia Tools and Applications*, vol.23, no.l, pp.7-30, May.2004. Article (CrossRef Link)

[3] I. Bartoli, G. Iacovoni, and F. Ubaldi, "A synchronization control scheme for videoconferencing services", *Journal of Multimedia,* vol.2, no.4, pp.1-9, Aug.2007. Article (CrossRef Link)

[4] S.S. Manvi, and P. Venkataram, "An agent based synchronization scheme for multimedia applications", *The Journal of Systems and Software*, vol.79, no.5, pp.701-713, May.2006. Article (CrossRef Link)

[5] F. Boronat Seguí, J. C. Guerri Cebollada, and J. Lloret Mauri, "An RTP/RTCP based approach for multimedia group and inter-stream synchronization", *Multimedia Tools and Applications*, vol.40, no.2, pp.285–319, Nov.2008. Article (CrossRef Link)

[6] S. E. Pomares Hernandez, L. A. Morales Rosales, J. Estudillo Ramirez, and G. Rodriguez Gomez, "Logical mapping: an intermedia synchronization model for multimedia distributed systems", *Journal of Multimedia*, vol.3, no.5, pp.33-41, Dec.2008.Article (CrossRef Link)

[7] C. Perkins, *RTP audio and video for Internet*, 1st Edition, Addison Wesley, USA, 2003. Article (CrossRef Link)

[8] A. Nafaa, T. Ahmed, Y. Hadjadj, and A. Mehaoua, "RTP4mux: a novel MPEG-4 RTP payload for multicast video communications over wireless IP", in *Proc. IEEE Packet Video (PV'03)*, Apr.2003.Article (CrossRef Link)

[9] K. H. Chi, L. H. Yen, C. Tseng, and T. L. Huang, "A causal multicast protocol for mobile distributed systems", *IEICE TRANS. INF. & SYST.*, vol. E83-D. no.12, pp.2065-2073, Dec.2000. Article (CrossRef Link)

[10] P. Chandra, and A. Kshemkalyani, "Causal multicast in mobile networks", in *Proc. of the IEEE Computer Societyapos 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications System*, pp.213-220, Oct .2004.

Article (CrossRef Link)

[11] R. Praskash, M. Raynal, and M. Singhal, "An adaptive causal ordering algorithm suited to mobile computing environments", *Journal of Parallel and Distributed Computing*, vol.41, no.2, pp.190-204, Mar.1997. Article (CrossRef Link)

[12] C. Benzaid, and N. Badache, "A causal multicast protocol for dynamic groups in cellular networks", in *Proc. of the 2008 Euro American Conference on Telematics and Information Systems*, pp.1-8, Sep.2008. Article (CrossRef Link)

[13] Z. J. Haas, and P. Agrawal, "Mobile-TCP: An asymmetric transport protocol design for mobile systems", in *Proc. of International Conference on Communications*, pp.1054-1058, Jun.1997. Article (CrossRef Link)

[14] E. Lopez, S. E. Pomares, G. Rodriguez, and Ma. Medina, "an efficient causal protocol with forward error correction for mobile distributed systems", *Journal of Computer Science*, vol.6, no.7, pp.756-768, Apr.2010. Article (CrossRef Link)

[15] L. Lamport, "Time clocks and the ordering of events in a distributed system", *Communications of the ACM*, vol.21, no.7, pp.558–565, Jul.1978. Article (CrossRef Link)

[16] S. E. Pomares Hernandez, J. Fanchon, and K. Drira, "the immediate dependency relation: an optimal way to ensure causal group communication", *Annual Review of Scalable Computing*, vol.6, Series on Scalable Computing. Ed. Y.C. Kwong, World Scientific, pp.61-79, 2004 Article (CrossRef Link)

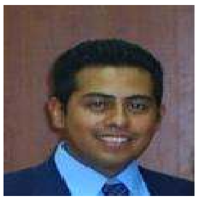
[17] J. Fanchon, K. Drira, and S. E. Pomares Hernandez, "Abstract channels as connectors for software components in group communication services", in *Proc. of Fifth Mexican International Conference on Computer Science* , pp.20–24, Sep.2004. Article (CrossRef Link)

[18] R. Baldoni, A. Mostefaoui, and M. Raynal, "Causal delivery of messages with real-time data in unreliable networks", *Journal of Real-Time System*, vol.10, no.3, pp.245-262, May.1996. Article (CrossRef Link)

[19] S. E. Pomares Hernandez, E. Lopez Dominguez, G. Rodriguez Gomez, and J. Fanchon, "An efficient Δ-causal algorithm for real time distributed system", *Journal of Applied Sciences*, vol.9, no.9, pp.1711-1718, 2009. Article (CrossRef Link)

[20] E. D. C. Group, *Sinalgo - simulator for network algorithms*, http://dcg.ethz.ch/projects/sinalgo/, 2008.

[21] D. Mackie, B. May and A. M. Franquet, *MPEG4 IP open source project*, http://mpeg4ip.sourceforge.net, 2000.



Eduardo Lopez Dominguez is a Researcher in the Department of Computer Science at National Laboratory of Applied Informatics (LANIA), in Veracruz, Mexico. He completed his PhD Degree at the National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico in 2010. Since 2004, he has been researching in the field of mobile distributed systems, partial order algorithms and multimedia synchronization.

Saul Eduardo Pomares Hernandez is a Researcher in the Computer Science Department at the National Institute of Astrophysics, Optics and Electronics (INAOE), in Puebla, Mexico. He completed his PhD Degree at the Laboratory for Analysis and Architecture of Systems of CNRS, France in 2002. Since 1998, he has been researching in the field of distributed systems, partial order algorithms and multimedia synchronization.

P. Gomez-Gil received the BSc degree from Universidad de las Americas A.C, Mexico and the MSc and PhD degrees from Texas Tech University, USA, all in computer science. She is currently a titular researcher in computer science at the National Institute of Astrophysics, Optics, and Electronics in Tonantzintla, Mexico. Her research interests include the design and use of artificial neural networks for solving complex problems related to forecasting, signal processing, pattern recognition and sensor-related applications. She is an active member of the IEEE and the ACM.

Jorge de la Calleja received his PhD in Computer Science in March 2008 from the National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico. His research interests include machine learning and computer vision. Since 2008, he is a Professor in the Polytechnic University of Puebla, Mexico, and lead the Group of Intelligent Computing and Systems.

Antonio Benitez is an associate professor of computer science at the Universidad Politéctica de Puebla (UPP). He holds MSc and PhD degrees in Computer Science from Universidad de las Américas Puebla (UDLAP) and a Engineering degree in Computer Science from Benemérita Universidad Autónoma de Puebla (BUAP). He has participated in projects related to the development of robotics and description of virtual environments. His current research areas include computer perception, distributed planning, reactive robotic and virtual reality. Dr. Benitez coordinates the graduate education department at the UPP

Antonio Marin-Hernandez, received the BSc degree in physics and the Master in Artificial Intelligence both the Universidad Veracruzana, Xalapa in 1995 and 1998. He received the ME and PhD by the INPT, Toulouse, in 2000 and 2004, working at the group Robotics and Artificial Intelligence from LAAS-CNRS, Toulouse. Since 1996 he is professor at the Department of Artificial Intelligence of the University of Veracruz and from 2005 a researcher.