

An Efficient Causal Protocol with Forward Error Correction for Mobile Distributed Systems

¹Eduardo Lopez Dominguez, ¹Saul E. Pomares Hernandez, ¹Gustavo Rodriguez Gomez,
²Maria Auxilio Medina and ³Jose A. Munoz Gomez

¹Department of Computer Science, National Institute of Astrophysics Optics and Electronics,
Luis Enrique Erro #1, 72840, Tonantzintla, Puebla, Mexico

²Department of Informatics, Polytechnic University of Puebla (UPP),
Third Rail Ejido Serrano s/n, 72640, San Mateo, Puebla, Mexico

³Department of Engineering, University of Guadalajara (CUCSUR),
Av. Independence National 151, 48900, Autlan de Navarro, Jalisco, Mexico

Abstract: Problem statement: Causal ordering is used in Mobile Distributed Systems (MDS) to reduce the non-determinism induced by four main aspects: host mobility, asynchronous execution, unpredictable communication delays and unreliable communication channels. Some causal protocols have been proposed for MDS. All of these protocols in order to ensure the causal order in unreliable channels use the method of Automatic Repeat Request (ARQ). They detect a lost message and carry out the retransmission of this message. This approach was not recommended in a real time mobile distributed system because it increased the transmission delay of the data and the overhead sent in the communication channels. **Approach:** In this study, we proposed a protocol that ensured the causal order of messages in unreliable and asynchronous MDS. In our protocol, the detection and recovery of lost messages was achieved by the method of Forward Error Correction (FEC) in a distributed form. One interesting aspect of our Causal-FEC protocol was that the redundant information sent in the wired and wireless communication channels is dynamically adapted to the behavior of the system. **Results:** Our protocol was efficient in terms of the overhead attached per message, the computational cost and the storage control information at a mobile host. **Conclusion:** The present study is one of the first works on causal algorithms based on forward error recovery in mobile networks.

Key words: Causal ordering, unreliable channels, causal distance, FEC

INTRODUCTION

The deployment of mobile distributed systems, in conjunction with wireless communication technologies and Internet, enables portable computing devices (referred in this study as mobile hosts), such as smart phones and Personal Digital Assistants (PDAs), to communicate anywhere and at anytime. The mobile distributed systems deal with new characteristics and constraints such as host mobility implying changeable physical network connections, limited processing and storage capabilities in mobile hosts compared with desktop computers and limited bandwidth on wireless communication channels.

For Mobile Distributed Systems (MDS), causal ordering algorithms are an essential tool to exchange information. The use of causal ordering provides built-

in message synchronization and reduces the non-determinism induced by four main aspects: host mobility, asynchronous execution, unpredictable communication delays and unreliable communication channels. Causal ordering provides an equivalent of the FIFO property at a global mobile multiparty communication level; it guarantees that actions, like requests or questions, are received before their corresponding reactions, results or responses. The concept of causal ordering has been of considerable interest to the design of mobile distributed systems and can be found in several domains, such as context-aware systems (Ghim *et al.*, 2007), ubiquitous agent systems (Tarkoma, 2003) and checkpoint protocols (Yi *et al.*, 2003).

Some works have been proposed to carry out a causal delivery of messages for mobile

Corresponding Author: Saul E. Pomares Hernandez, Department of Computer Science,

National Institute of Astrophysics, Optics and Electronics, Luis Enrique Erro #1, 72840, Tonantzintla, Puebla, Mexico Tel: (01) (222) 266 31 00/Ext.8227 Fax: 01(222) 266 31 52

distributed systems (Alagar and Venkatesan, 1997; Prakash *et al.*, 1996; Kuang *et al.*, 2000; Chandra and Kshemkalyani, 2004; Benzaid and Badache, 2005; Lopez *et al.*, 2008). Most of these causal protocols consider reliable communication channels. However, the loss of messages and the variable latency are inherent characteristics of the wireless communication channels. Some works, in order to tolerate the loss of messages and ensure the causal ordering, use a kind of Automatic Repeat Request method (ARQ) (Acharya and Badrinath, 1996; Alberto and Francesco, 2001; Anastasi *et al.*, 2004). They detect a lost message and carry out the retransmission of this message. This approach is suitable for applications that broadcast discrete data such as images, text and binary files, which are data that tolerate the delay, but are sensitive to the loss of messages. On the other hand, there are applications that broadcast real time discrete and continuous data (e.g., audio and video), which have different transmission characteristics, such as time constrains. Hence, the retransmission of messages is not suitable for such applications because this increment the transmission delay. For this kind of environments a Forward Error Correction scheme (FEC) is preferable over a backward recovery scheme (Perkins, 2003).

In general the protocols that use a FEC approach introduce certain information redundancy in order to identify and recover the lost data. The redundancy in causal protocols represents the number of times that information about a causal message is sent in the system. In this study, we propose a causal protocol that ensures the causal order of messages in unreliable and asynchronous MDS. In our protocol, the detection and recovery of lost messages is achieved by the approach of forward error correction proposed in (Pomares *et al.*, 2009) for non mobile constrained systems. We apply this FEC scheme to the causal protocol MOCABI presented in (Lopez *et al.*, 2008) in order to support mobile distributed systems over unreliable wireless channels. MOCABI ensures the causal delivery of messages according to the causal view of the mobile hosts. The protocol MOCABI is based on the Immediate Dependency Relation (IDR) relation (Pomares *et al.*, 2004). The IDR relation identifies the necessary and sufficient control information to be attached to each message to ensure the causal order in a reliable network. MOCABI, at the intra-base communication level (wireless connection), only sends as causal overhead timestamped per message, a structure of bits $h(m)$, where each bit in the $h(m)$ is determined according to the IDR. In order to support the loss of messages, we introduce redundancy on the control information attached to the message sent at

wireless and wired communication channels. The redundant control information is calculated based on the causal distance (Definition 3) between messages. Instead of restricting the causal information to its immediate predecessors, messages that are IDR related have a causal distance of one (i.e., no intermediate causal message exists between them). The causal information attached to a message concerns messages in its causal past up to a maximal causal distance (greater than one).

One interesting aspect of our Causal-FEC protocol is that the redundant information sent in wireless communication channels is composed by a set of bits, which is dynamically determined according to the behavior of the system and the predetermined causal distance. This redundant information greatly increase the probability that causal order delivery will be obtained, even in the presence of lost messages and significant network delays, by moderately increasing the overhead in the wireless channels. Our causal protocol is efficient in terms of the overhead attached to messages, the computational cost and storage control information on the mobile hosts. The present study is one of the first works on causal algorithms oriented to the forward error recovery over mobile networks.

MATERIALS AND METHODS

Preliminaries:

The system model: We consider in this study that an MDS runs over a wireless network infrastructure which consists of two kinds of entities: Base Station (BS) and Mobile Host (MH). A BS has the necessary infrastructure to support and communicate with mobile hosts. The BS communicates with mobile hosts through wireless communication channels. The area covered by a BS is called a cell (Fig. 1).

An MH is a host that can move while retaining its network connection. At any given time, an MH is assumed to be within the cell of at most one BS, which is called its local BS. An MH can communicate with other MHs and BSs only through its local BS.

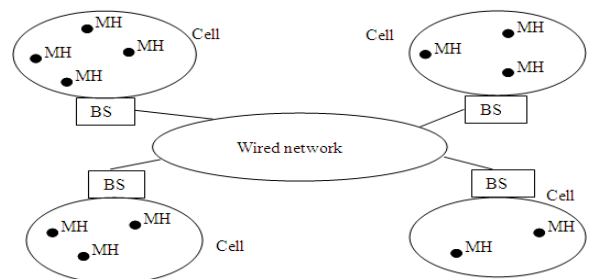


Fig. 1: Physical architecture of a MDS

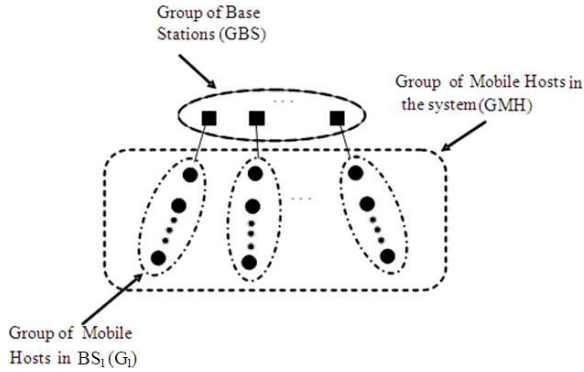


Fig. 2: Logical structure considered for a mobile distributed system

We assume that the wireless communication channels between MHs and BSs are unreliable and asynchronous. The base stations are connected among themselves using wired channels. The BSs and the wired channels constitute the static network. We assume that the wired channels are unreliable (the messages can be lost during its broadcast), with an arbitrary but finite amount of time to deliver messages. Due to system asynchrony and unpredictable communication delays, the messages on an MDS from MH-MH can arrive in a different order as they were sent.

On the other hand, from a logical point of view, we consider that the entities of the MDS are structured into two main communication groups, one conformed by the base stations ($GBS = \{BS_1, BS_2, \dots, BS_s\}$) and the other integrated by mobile hosts ($GMH = \{p_1, p_2, \dots, p_n\}$), where s and n are the number of base stations and mobile hosts, respectively, in the mobile distributed system. The GMH is subdivided into subgroups (G_i), one for each BS (Fig. 2).

The BSs in the GBS and the mobile hosts in a G_i communicate by unreliable asynchronous message passing. We consider a finite set of messages M with $m \in M$, identified by a tuple $m = (p, t)$, where p is the sender mobile host, such that $p \in GMH$ and t is the logical clock for messages of p when m is sent. When we need to refer to a specific process with its respective identifier, we write p_i . The set of destinations of a message m is always GMH.

Background and definitions: Causal ordering delivery is based on the causal precedence relation defined by (Lamport, 1978). The happened-before relation establishes over a set of events the possible precedence dependencies without using physical clocks. It is a partial order defined as follows:

Definition 1: The causal relation “ \rightarrow ” is the least partial order relation on a set of events satisfying the following properties:

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

By using Definition 1 we say that a pair of events are concurrently related “ $a \parallel b$ ” only if $\neg(a \rightarrow b \vee b \rightarrow a)$.

The precedence relation on messages denoted by $m \rightarrow m'$ is induced by the precedence relation of events and is defined by:

$$m \rightarrow m' \Leftrightarrow \text{send}(m) \rightarrow \text{send}(m')$$

The immediate dependency relation: The Immediate Dependency Relation (IDR) formalized by (Pomares *et al.*, 2004) is the propagation threshold of the control information regarding the messages sent in the causal past that must be transmitted to ensure a causal delivery. We denote it by \downarrow and its formal definition is the following:

Definition 2: Immediate Dependency Relation “ \downarrow ” (IDR):

$$m \downarrow m' \Leftrightarrow [(m \rightarrow m') \wedge \forall m'' \in M, \neg(m \rightarrow m'' \rightarrow m')]$$

Thus, a message m directly precedes a message m' , if and only if no other message m'' belonging to M exists (M is the set of messages of the system), such that m'' belongs at the same time to the causal future of m and to the causal past of m' .

The IDR is the transitive reduction of the Happened-before relation (Definition 1). This relationship is important because if the delivery of messages respects the order of transmission for all pairs of messages in an IDR, then the delivery of messages respects the causal order for all processes that belong to the group of destinations ($p \in GMH$). This property is formally defined for the broadcast case as follows:

Property 1:

$$\begin{aligned} &\text{If } m, m' \in M, m \downarrow m' \Rightarrow \\ &\forall p \in GMH: \text{delivery}(p, m) \rightarrow \text{deliver}(p, m) \text{ then} \\ &m \rightarrow m' \Rightarrow p \in GMH: \text{delivery}(p, m) \rightarrow \text{delivery}(p, m') \end{aligned}$$

Causal information that includes the messages immediately preceding a given message is sufficient to ensure a causal delivery of such message.

The causal distance: The causal distance identifies the number of causal messages that exist in a linearization between a pair of messages in the system (Lopez *et al.*, 2005). Formally, the causal distance is defined as follows:

Definition 3: Let m and m' be messages, the distance $d(m,m')$ is defined for any pair m and m' ($\text{send}(m) \rightarrow \text{send}(m')$), such that $d(m,m')$ is the integer n for some sequence of messages $(m_i, i = 0 \dots n)$, with $m = m_0$ and $m' = m_n$, such that $\text{send}(m_i) \downarrow \text{send}(m_{i+1})$ for all $i = 0 \dots n-1$.

The causal-FEC protocol:

The FEC mechanism: All FEC protocols introduce some kind of redundancy to support the loss of information. The redundancy in causal protocols represents the number of times that information about a causal message is sent in the system. The causal protocol MOCIVI presented in (Lopez *et al.*, 2008), which uses the IDR relation, is efficient because the IDR relation identifies the necessary and sufficient control information that needs to be attached to each message sent over the mobile system. In this protocol, each mobile host p uses a structure of bits in order to establish an immediate dependency relation (Definition 2) among messages. The content of structure of bits at a mobile host p is the only control information attached per message in the wireless channel. Each bit in this structure identifies a causal message m that has a potential IDR with the next message to be sent by p . In this protocol, through the control information stored at the base station and the structure of bits attached to the messages sent in the system, the base stations can determine the immediate dependency relation between messages sent by MHs on different BSs.

Even when this is an efficient protocol, redundant control information is still transmitted in some communication scenarios. Our causal-FEC protocol identifies and uses this inherent redundancy and only adds extra redundancy when it is needed. The purpose of adding extra redundancy is to increase the probability that causal order delivery will be obtained, even in the presence of lost messages and significant network delays.

Analysis of the redundancy and IDR dependencies in mobile environments:

To ensure causal ordering based on the IDR we only need sends control information attached to each message about messages with an immediate dependency relation. For two messages that are IDR-related ($m \downarrow m'$), the causal distance is equal to one ($d(m,m') = 1$). Note that for the serial case, a message m has only one immediate predecessor (best case) and that a message m can have at most n immediate predecessor messages, one for each process.

For the serial case, in which the messages are IDR-related, there is no redundancy in the control information sent. For example, in the serial scenario depicted in Fig. 3, message m_3 only sends information about message m_2 and message m_2 only sends information about message m_1 . In this case, if a message is lost, the causal order delivery can be violated. As shown in Fig. 3, the causal order delivery is violated because at the reception of message m_3 , the base station BS_1 cannot determine if a message preceding m_2 exists or not. With the IDR information on m_3 , base station BS_1 can only detect that it missed message m_2 . In order not to stop the system execution, BS_1 considers message m_2 as lost and then delivers m_3 to mobile host p_1 . In this scenario, m_1 can be delivered after m_3 to mobile host p_1 , which violates the causal ordering.

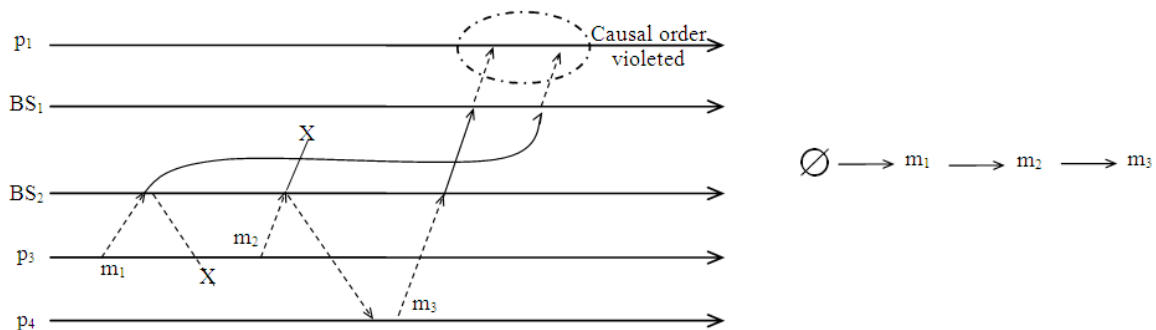


Fig. 3: Example scenario and its associated IDR graph

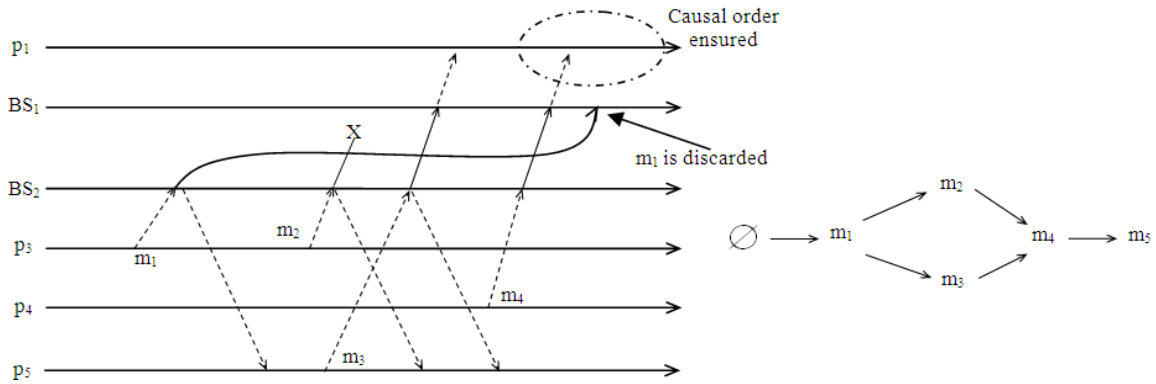


Fig. 4: Example scenario and its associated IDR graph

For the concurrent relation, inherent redundancy exists on the control information sent. For example, in the scenario depicted in Fig. 4, messages m_2 and m_3 have the same immediate predecessor m_1 and therefore, m_2 and m_3 send information about message m_1 . If either message m_2 or m_3 is lost, message m_1 can still be detected as shown in Fig. 4. In this scenario, m_2 is lost and m_3 successfully arrives at BS_1 . With the IDR information on m_3 , base station BS_1 determines that m_1 exists, which precedes message m_3 . To deliver m_3 , base station BS_1 establishes message m_1 as lost. In this scenario, m_1 arrives at BS_1 after the delivery of message m_4 , but since message m_1 has been established as lost, it is immediately discarded. Therefore, causal order is ensured at mobile host p_1 .

The FEC's propagation constraints for causal ordering: In order to support the loss of messages, Pomares *et al.* (2009) propose to increase the redundancy in the control information sent per message by sending information about causally-related messages with a causal distance greater than one. For example, in Fig. 3, if a causal distance of two ($\text{causal_distance} = 2$) is considered, this means that message m_3 must send information about m_2 and m_1 .

According to this approach, in order to be efficient, the redundancy must be increased considering the inherent redundancy introduced by the IDR relation. Formally, in our context the redundancy about a message m , denoted by $\text{redundancy}_p(m)$, determines the number of times that the information about a causal message m has been seen (received) by a mobile host p . As previously described, the redundancy increases as the number of concurrent messages increases. Taking into account $\text{redundancy}_p(m)$ with a causal distance greater than one ($\text{causal_distance} > 1$), it was established that a message m' must include information about a

causal message m ($m \rightarrow m'$) only if the following propagation constraints are satisfied:

- PC1: $d(m, m') \leq \text{causal_distance}$ and
- PC2: $\text{causal_distance} > \text{redundancy}_p(m)$

With both of these PCs, the control information sent per message is dynamically adapted to the behavior of the system by only introducing redundancy when it is needed. For example, with $\text{causal_distance} = 2$, message m_3 , shown in Fig. 3, must send causal information about m_2 and m_1 because $\text{redundancy}_4(m_1)$ equal to one and a causal distance of $d(m_1, m_3) = 2$ and $d(m_2, m_3) = 1$, respectively. Nevertheless, for the scenario presented in Fig. 4, message m_4 must send information only about messages m_2 and m_3 and not about m_1 , even when $d(m_1, m_4) = 2$. This is done because the $\text{redundancy}_4(m_1)$ is equal to 2 and therefore, it does not satisfy the second PC.

In a general case, according to the analysis presented in (Perkins, 2003), it is sufficient to take a causal_distance equal to 5 since the probability that three or more consecutive and concurrent messages can be lost is very low.

RESULTS

The algorithm: In our study, we consider that messages can be lost during their transmission over the wireless or wired communication channels, which in some scenarios can cause the causal order to be violated, (Fig. 3). In order to ensure the causal delivery in the mobile distributed system, we increase the redundancy in the sent control information per message by sending information about causally related messages with a causal distance greater than one.

Each time that a message m is received by a base station BS or a mobile host p , we take two corrective actions in order to ensure the causal order at unreliable communication channels. First, we delay the delivery of messages when they do not satisfy the causal dependencies and secondly, we discard messages when their deadlines are exceeded. The deadline of a message m , denoted by $deadline(m)$, can be determined through a centralized method (Baldoni *et al.*, 1996) or a distributed method (Pomares *et al.*, 2009). Here, we do not present how the $deadline(m)$ is calculated, you can refer to these works for details.

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

- $mes_received(p)$ is a counter that is increased each time a message is received by the mobile host p
- $mes_sent(p)$ is a counter which is incremented each time a message is sent by the mobile host p
- $\Phi(p)$ is a structure of bits composed by elements $bit_T = 1b$. Each bit_T in $\Phi(p)$ identifies a message m' in the causal past of p that satisfies, with respect to m , the PC1 and PC2 propagation constraints. The size of $\Phi(p)$ fluctuates between $1 \leq |\Phi(p)| \leq n$, where n is the number of mobile hosts in the mobile distributed system

The structures and variables stored at the mobile host p_i are initialized with:

- $\Phi(p_i) \leftarrow \emptyset$
- $bit_T \leftarrow 1b$
- $bit_F \leftarrow 0b$
- $mes_received(p_i) = 0$
- $mes_sent(p_i) = 0$

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

- $mes_sent(BS)$ is a counter, which is incremented each time a message is sent by the base station BS in its cell
- $VT(BS)$ 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 . The size of VT is equal to the number of mobile hosts in the group. $VT(BS)$ contains the local view that a mobile host $p \in BS$ has of the messages sent in the system. An element $VT(BS)[i]$ represents the most recently sent message by p_i and 'seen' in causal order by the BS
- $CI(BS)$ is the control information structure. It is a set of entries (i, t, k, DC) . The entry (i, t, k, DC)

represents a message sent by the mobile host p_i with logical local clock $t = VT(BS)[i]$ and $k = mes_sent(BS)$, with a causal distance equal to DC .

The structures and variables stored at the base station BS_r are initialized with:

- $VT(BS_r)[i] = 0 \quad \forall i:1..n;$
- $CI(BS_r) \leftarrow \emptyset$
- $H(m) \leftarrow \emptyset$
- $mes_sent(BS_r) = 0$

The following message structures are used in the MDS by the mobile hosts and base stations (Fig. 5):

- The messages sent in the wireless communication channels by mobile hosts to its base station are identified by m and have the following form: $m \equiv (i, t, mes_received(p), data, h(m))$, where the structures i, t and $mes_received(p)$ have been previously described and:
 - $h(m)$ is a structure of bits, which is created at the moment of transmission of a message m by mobile host p_i . The $h(m)$ structure is composed by elements $bit_T = 1b$, where bit_T identifies a message m' that satisfies the PC1 and PC2 propagation constraints with respect to m
- A message m sent among base stations BSs is denoted by $bs(m)$ and it is composed by a quintuplet $bs(m) \equiv (i, t, data, H(m))$, where:
 - $data$ is the content of the message and
 - $H(m)$ is composed of a set of elements (i, t) , which represent messages that satisfy, with respect to m , the PC1 and PC2 propagation constraints. A structure $H(m)$ is created at the moment that a broadcast message is sent by a base station
- A message m received by a BS_1 from a mobile host $p \in G_1$ and re-sent by such BS_1 in its cell, consists of a quintuplet that we call $intra(m) \equiv (i, t, data, h'(m))$
 - $h'(m)$ is a structure of bits, which is created at the moment of transmission of a message m by base station BS . This structure is composed by binary elements $bit_T = 1b$ and $bit_F = 0b$, where bit_T identifies a message m' that satisfies PC1 and PC2 with respect to m and that the BS_1 has not ensured its causal delivery to its members and bit_F identifies a message m' that does not satisfy PC1 or PC2
- A message $bs(m)$ received by a BS_1 and re-sent within its cell, consists of a quintuplet that we call $inter(m) \equiv (i, t, data, h'(m))$

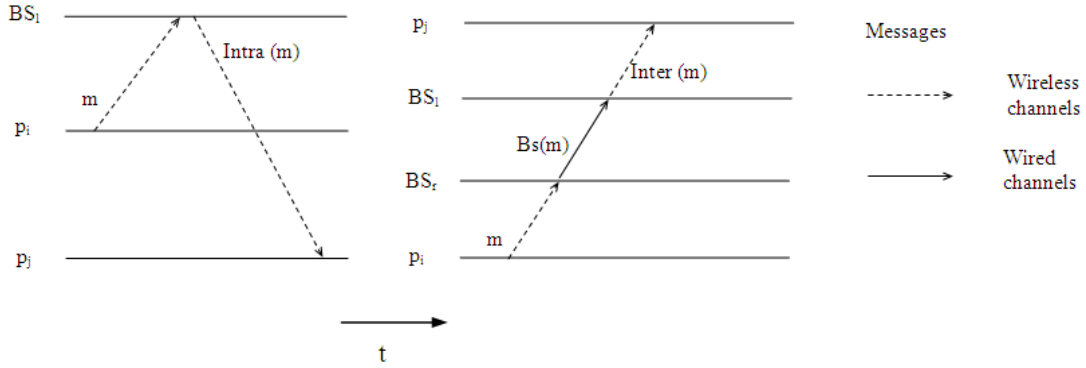


Fig. 5: Messages sent at the wired and wireless communication channels, respectively

Table 1: Global functions at base station

1.	let update_ h'(m)_and_CI(k, t, d, DC) {
2.	if (DC = causal_distance) then
3.	h'(m) ← h'(m) ∪ bit_F
4.	CI(BSr) ← CI(BSr) / (k, t, d, DC)
5.	else
6.	h'(m) ← h'(m) ∪ bit_T endif
7.	Return h'(m) }

Table 2: Diffusion of message m by a mobile host p_i

1.	mes_sent(p) = mes_sent(p) + 1
2.	h(m) ← Φ(p _i)
3.	m = (i = id_host, t = mes_sent(p), mes_received(p), data, h(m))
4.	Diffusion: send(m) /* sending of message m to local BS*/

Table 3: Reception of message intra(m) or inter(m) by a mobile host p_i, i ≠ j

	/* intra(m) = (i, t, data, h'(m)) or inter(m) = (i, t, data, h'(m)) */
1.	if (t < mes_received(p _i)) then
2.	discard (intra(m) inter(m))
3.	else
4.	wait ((t = mes_received(p _i) + 1) or (deadline(intra(m') inter(m') = (i, t')) has expired such that t' < t))
5.	delivery(intra(m) inter(m))
6.	mes_lost(p _i) = t - mes_received(p _i)
7.	while (mes_lost(p _i) > 1) do
8.	Φ(p _i) ← Φ(p _i) ∪ bit_T
9.	mes_lost(p _i) = mes_lost(p _i) - 1
10.	Endwhile
11.	mes_received(p _i) = t
12.	Φ(p _i) ← Φ(p _i) ∪ bit_T
13.	∀ { bit_F } ∈ h'(m)
14.	Φ(p _i) ← Φ(p _i) / bit_T endif

Protocol description: In the next paragraphs, we describe the steps that are carried out during the diffusion and reception of a message by mobile hosts and base stations in order to ensure the causal order at unreliable wireless and wired communication channels. The protocol specification is given in Table 1-5.

Diffusion of message m by a mobile host p: The emission of a message $m \equiv (i, t, \text{mes_received}(p), \text{data}, h(m))$ by mobile host p to its local base station BS is carried out in four steps. First, the counter $\text{mes_sent}(p)$ is incremented by one (Line 1, Table 2). This counter keeps

the number of messages sent by mobile host p to local base station BS. Second, the mobile host p creates the structure of bits $h(m)$, which is a copy of structure of bits $\Phi(p)$, Line 2, Table 2. Each bit_T in $h(m)$ identifies a message that satisfies, with respect to m, the PC1 and PC2 propagation constraints. Next, the mobile host p attaches $h(m)$ to the message m (Line 3, Table 2). In the following, we will describe how the redundancy in the structure of bits $h(m)$ allow us to support the loss of messages in the wireless communications channels. Finally, the diffusion of message m to local base station BS is done by mobile host p, Line 4, Table 2.

Table 4: Reception-sending of message $m = (i, t, \text{mes_received}(p_i), \text{data}, h(m))$ by the base station BS_r

1.	if $i \in BS_r$ then
2.	if $(t < VT(BS_r)[i])$ then
3.	discard(m)
4.	else
5.	wait_FIFO $((t = VT(BS_r)[i] + 1)$ or $(\text{deadline}(m' = (i, t'))$ has expired such that $t' < t)$)
6.	delivery(m)
7.	$VT(BS_r)[i] = t$
8.	$\forall \{ \text{bit}_T \} \in h(m)$
9.	if $\exists (k, t, d, DC) \in CI(BS_r) \mid d = \text{mes_received}(p_i)$ then
10.	$(k, t, d, DC) \leftarrow (k, t, d, DC + 1)$
11.	$H(m) \leftarrow H(m) \cup (k, t)$
12.	update_h'(m)_and_CI $((k, t, d, DC))$ endif
13.	$\text{mes_received}(p_i) = \text{mes_received}(p_i) - 1$
14.	$\forall (x, y) \in H(m)$
15.	if $\exists (k, t) \in H(m) \mid k = x$
16.	$H(m) \leftarrow \max((x, y), (k, t))$ endif
17.	intra(m) $\equiv (i, t = \text{mes_sent}(BS_r) + 1, \text{data}, h'(m))$
18.	bs(m) $\equiv (i, t, \text{data}, H(m))$
19.	Diffusion : send intra(m) /* sending of message intra(m) to local mobile hosts */
20.	Diffusion : send bs(m) /* sending of message bs(m) to other base stations*/
21.	endif // line 5
22.	$\text{mes_sent}(BS_r) = \text{mes_sent}(BS_r) + 1$
23.	$CI(BS_r) \leftarrow CI(BS_r) \cup \{ (i, t, \text{mes_sent}(BS_r), DC = 0) \}$

Table 5: Reception-send of message $bs(m) \equiv (i, t, \text{data}, H(m))$ by a base station BS_r .

1.	if $i \notin BS_r$ then
2.	if $(t < VT(BS_r)[i])$ then
3.	discard (bs(m))
4.	else
5.	wait $((t = VT(BS_r)[i] + 1)$ or $(\text{deadline}(m' = (i, t'))$ has expired such that $t' < t)$ and $(\forall m' = (s, x) \in H(m) : x \leq VT(BS_r)[s]$ or $\text{deadline}(m')$ has expired))
6.	delivery(bs(m))
7.	$\forall (s, x) \in H(m)$
8.	if $(x > VT(BS_r)[s])$ then
9.	$VT(BS_r)[s] = x$ endif
10.	$VT(BS_r)[i] = t$
11.	$\forall (x, y) \in H(m)$ // construction the bits vector
12.	if $\exists (k, t, d, DC) \in CI(BS_r) \mid x = k$ and $t \leq y$ then
13.	$(k, t, d, DC) \leftarrow (k, t, d, DC + 1)$
14.	update_h'(m)_and_CI $((k, t, d, DC))$ endif
15.	inter(m) $\equiv (i, t = \text{mes_sent}(BS_r) + 1, \text{data}, h'(m))$
16.	Diffusion : send(inter(m)) /* sending of message inter(m) to local mobile hosts */
17.	endif
18.	$\text{mes_sent}(BS_r) = \text{mes_sent}(BS_r) + 1$
19.	$CI(BS_r) \leftarrow CI(BS_r) \cup \{ (i, t, \text{mes_sent}(BS_r), DC = 0) \}$

Reception-sending of message $m = (i, t, \text{mes_received}(p_i), \text{data}, h(m))$ by a base station BS_r :

When a message $m \equiv (i, t, \text{mes_received}(p_i), \text{data}, h(m))$ is received at the local base station BS_r , it will be immediately discarded if it has already been marked as lost $(t < VT(BS_r)[i])$, Lines 2-3, Table 4. If m is not discarded, it is delivered as soon as the FIFO condition becomes true (Lines 5-6, Table 4). The condition FIFO ensures that a message m is delivered until all messages sent before it by the mobile host p_i , identified by i , have been delivered at BS_r $(t = VT(BS_r)[i] + 1)$ or has been established as missing, i.e. their lifetime have expired.

A posteriori, these messages are marked as lost in Line 7, Table 4 and therefore, they will never be delivered. When the FIFO condition becomes true, the message m is delivered to base station BS_r and the $VT(BS_r)$ vector is updated into the position that identifies to mobile host p_i with logical clock t . Afterwards, the base station BS_r sends message m to the mobile hosts in its cell and to the other base stations in the mobile distributed system. This is done through the diffusion of messages intra(m) and bs(m) by BS_r , respectively.

The message $\text{intra}(m) \equiv (i, t, \text{data}, h'(m))$ is constructed by BS_r as follows. First, the base station BS_r

forms the structure of bits $h'(m)$, which is created at the moment of transmission of the message m by base station BS to its local mobile hosts, Lines 8-13, Table 4 and Lines 1-7, Table 1. In our study, $h'(m)$ is composed by binary elements $bit_T = 1b$ and $bit_F = 0b$, where bit_T identifies a message m' that satisfies, with respect to m , the PC1 and PC2 propagation constraints and bit_F identifies a message m' that does not satisfy PC1 or PC2. In this case, it is not necessary to send the control information about m' over wireless and wired channels.

On the other hand, the messages $bs(m) \equiv (i, t, data, H(m))$ sent by BS to other base stations are constructed as follows. According to the cardinality of the $h(m)$ structure ($0 \leq |h(m)| \leq n-1$), the base station BS creates the structure $H(m)$ by adding entries from the $CI(p)$ (Lines 8-11, Table 4) to it. The $H(m)$ structure is composed of a set of elements (i, t) , which represent messages that satisfy the propagation constraints (PC1 and PC2) with respect to m .

Reception-sending of message $bs(m) \equiv (i, t, data, H(m))$ by a base station BS: When a message $bs(m) \equiv (i, t, data, H(m))$ is received by a base station BS, the BS verifies if $bs(m)$ has been marked as lost ($t < VT(BS)[i]$). When this condition becomes true, the message is discarded, Line 3, Table 5; otherwise, the message $bs(m)$ is delivered as soon as the FIFO and causal delivery condition is satisfied. This causal delivery condition ensures that a message $bs(m)$ will be delivered if and only if all messages causally related to it have either been delivered or have been established as missing, i.e. their lifetime has expired. A posteriori, these messages are marked as lost in Lines 7-10, Table 5 and therefore, they will never be delivered.

When the causal delivery condition is satisfied, the message $bs(m)$ is delivered to base station BS, line 6, Table 5. Afterwards, BS carries out the sending of message $bs(m)$ to the mobile hosts located within its cell. This is done through the diffusion of message $inter(m) \equiv (i, t, data, h'(m))$. In this case, the structure of bits $h'(m)$ attached to message $inter(m)$ is constructed by BS as follows; Line 11, Table 5. For each element $m' = (x, y)$ that belongs to $H(m)$, the base station verifies if an element $(k, t, d, DC) \in CI(BS_r)$ exists such that $x = k$ and $t \leq y$. When this condition becomes true, the BS increments by one the causal distance between m' and m , Line 13, Table 5. Later on, if the distance causal between m' and m is equal to the predetermined causal distance ($DC = causal_distance$), the base station BS deletes the element (k, t, d, DC) that belongs to $CI(BS_r)$ and attaches a bit_F to $h'(m)$; otherwise, the base station BS attaches a bit_T to the

structure of bits $h'(m)$, Lines 1-7, Table 1. In our study, each bit_F identifies a message m' that does not satisfy the PC1 and PC2 propagation constraints.

Reception of message $intra(m)$ or $inter(m)$ by a mobile host p : When a message $intra(m) = (i, t, data, h'(m))$ or $inter(m) = (i, t, data, h'(m))$ is received at a mobile host p , it will be immediately discarded if it has already been marked as lost ($t < mes_received(p_i)$), Line 1-2, Table 3. If $intra(m)$ or $inter(m)$ is not discarded, it will be delivered as soon as the FIFO delivery condition becomes true (Line 4-5, Table 3). This delivery condition ensures that a message $intra(m)$ or $inter(m)$ will be delivered if and only if all messages causally related to it have either been delivered or have been established as missing. When the messages that causally precede message $intra(m)$ or $inter(m)$ have been established as missing, the mobile host p carries out the following FEC mechanism. First, the FEC mechanism calculates the amount of messages established as missing, Line 6, Table 3. Later on, in order to ensure the causal delivery, the mobile host p attaches an element bit_T to the structure of bits $\Phi(p)$ by each message marked as missing, Lines 7-10, Table 3. Finally, in order to delete the bit_T that identifies a message m' with a causal distance greater than the predetermined causal distance with respect to m , the mobile host p updates its structure of bits $\Phi(p)$ with the $h'(m)$ structure attached to message $intra(m)$ or $inter(m)$. This is done by our protocol as follows, Lines 12-14, Table 3. For each element bit_F that belongs to $h'(m)$, the mobile host p must delete an element bit_T of the structure of bits $\Phi(p)$. In this case, bit_F identifies a message m' that does not satisfy the PC1 propagation constraint and therefore, it is not necessary to send the control information about m' . Later on, due to the delivery of message $intra(m)$ or $inter(m)$, the mobile host p attaches a bit_T to the structure of bits $\Phi(p)$, line 12, Table 3.

Correctness proof: To show that our protocol ensures the causal delivery (correctness), we provide a correctness proof. In order to do the proof as simple as possible, we divide the correctness proof into two parts: first, we focus on the novel part for the unreliable wireless channels, which is the redundancy control information (bits) attached to the sent messages at wireless channels and the causal information stored at the base stations (Theorem 1). Secondly, we show that with this redundancy information, it is possible to ensure the causal order in mobile distributed systems with unreliable communication channels (Theorem 2).

Consider two messages $m_k = (p_i, a, \text{event}, h(m_k))$ and $m_l = (p_j, b, \text{event}, h(m_l))$, where p_i and p_j are the sender mobile hosts of m_k and m_l , respectively, a and b are the sequential ordered logical clocks for messages of p_i and p_j when m_k and m_l are sent, respectively. And finally $h(m_k)$ and $h(m_l)$ are the structures of bits when the messages m_k and m_l are sent, respectively.

Theorem 1: $\forall \text{bit_T}_k \in h(m_l) \exists (x, y, k') \in \text{CI}(\text{BS})$ such that $k = k'$, where (x, y) identifies a message m_k , which has $d(m_k, m_l) \leq \text{causal_distance}$.

Main steps of the proof: The proof is composed by two lemmas and a proposition. The lemmas are intermediate results necessary for our proof:

- Lemma 1 shows that if bit_T_k belongs to the causal history of a message m_l , then the message identified by bit_T_k causally precedes message m_l
- Lemma 2 indicates that the message m_k satisfies PC1 and PC2 propagation constraints with the other message m_l if and only if the bit_T_k belongs to the causal history of the message m_l
- Proposition 1 shows that through the bits structure ($h(m)$) attached to the sent message m_l and the causal information at the base stations ($\text{CI}(\text{BS})$), we identify messages that satisfy the PC1 and PC2 propagation constraints with respect to m_l (Theorem 1)

Lemma 1:

$$\text{bit_T}_k \in h(m_l) \Rightarrow m_k \rightarrow m_l$$

Proof: By line 2, Table 2, we have that $\text{bit_T}_k \in h(m_l)$ if and only if $\text{bit_T}_k \in \Phi(p_j)$ when the sending of m_l is carried out by p_j . We denote it by $\text{send}(p_j, m_l)$. By using line 12, Table 3, we have that $\text{bit_T}_k \in \Phi(p_j)$ only after the delivery $m_k = (p_i, a, \text{event}, h(m_k))$ at p_j . This implies that the delivery of m_k precedes the sending of m_l ($\text{delivery}(p_j, m_k) \rightarrow \text{send}(p_j, m_l)$). Therefore, $m_k \rightarrow m_l$. \diamond

Lemma 2:

$$d(m_k, m_l) \leq \text{causal_distance} \Rightarrow \text{bit_T}_k \in h(m_l)$$

The proof is divided into two steps: First, we show that $d(m_k, m_l) \leq \text{causal_distance} \Rightarrow \text{bit_T}_k \in h(m_l)$ and second, we show that $\text{bit_T}_k \in h(m_l) \Rightarrow d(m_k, m_l) \leq \text{causal_distance}$

Step 1:

$$d(m_k, m_l) \leq \text{causal_distance} \Leftrightarrow \text{bit_T}_k \in h(m_l)$$

The proof is by contrapositive. We proof that $\text{bit_T}_k \notin h(m_l) \Rightarrow \exists m_r, m_{r+1}, m_{r+2}, \dots, m_{n-1}, m_n \in M$ such that $m_k \downarrow m_r \downarrow m_{r+1} \downarrow m_{r+2}, \dots, \downarrow m_{n-1} \downarrow m_n \downarrow m_l$; thus, the message m_k with respect to message m_l has a causal distance greater to the defined causal distance ($d(m_k, m_l) > \text{causal_distance}$). Let $\text{bit_T}_k \notin h(m_l)$, only one event can delete bit_T_k of $\Phi(p_j)$ before sending m_l ($\text{send}(p_j, m_l)$), this is:

By lines 13 and 14, Table 3, bit_T_k is removed from $\Phi(p_j)$ when the delivery of message m_n is carried out with $\text{bit_F}_k \in h(m_n)$ at p_j ($\text{delivery}(p_j, m_n)$). The binary element bit_F_k is attached to $h(m_n)$ if and only if the $d(m_k, m_n) = \text{causal_distance}$, Lines 2 and 3, Table 1. This implies that for pair m_k and m_n a sequence of messages exists such that $\text{send}(m_k) \downarrow \text{send}(m_r) \downarrow \text{send}(m_{r+1}) \downarrow \text{send}(m_{r+2}) \downarrow \dots \downarrow \text{send}(m_{n-1}) \downarrow \text{send}(m_n)$, where the causal distance between m_k and m_n is equal to the defined causal distance. Thus, the $\text{delivery}(p_j, m_n)$ at mobile host p_j before the sending of m_l deletes the bit_T_k of $\Phi(p_j)$. Therefore, the $d(m_k, m_l) > \text{causal_distance}$ because $\text{bit_T}_k \notin h(m_l)$.

If this event does not occur, we have that $\text{bit_T}_k \in \Phi(p_j)$ when the $\text{send}(p_j, m_l)$ is carried out and by Line 2, Table 2, we have that $\text{bit_T}_k \in h(m_l)$. \diamond

Step 2:

$$\text{bit_T}_k \in h(m_l) \Rightarrow d(m_k, m_l) \leq \text{causal_distance}$$

The proof is by contradiction. By lemma 1, we know that if $\text{bit_T}_k \in h(m_l)$, then $m_k \rightarrow m_l$ with $p_i \neq p_j$. We suppose that there are any messages $m_r, m_{r+1}, m_{r+2}, \dots, m_{n-1}, m_n$ such that $\text{send}(p_i, m_k) \rightarrow \text{send}(p_r, m_r) \rightarrow \text{send}(p_r, m_{r+1}) \rightarrow \text{send}(p_j, m_{r+2}) \rightarrow \dots \rightarrow \text{send}(p_j, m_{n-1}) \rightarrow \text{send}(p_j, m_n) \rightarrow \text{send}(p_j, m_l)$ and in addition that $d(m_k, m_n) = \text{causal_distance}$. The proof considers the following case:

- The delivery m_k causally precedes to m_n ($\text{delivery}(p_j, m_k) \rightarrow \text{delivery}(p_j, m_n)$) at p_j . By the step 1, we know that $\text{bit_F}_k \in h(m_n)$. Hence, on the delivery m_n at mobile host p_j , bit_T_k is deleted by Lines 13 and 14, Table 3. When performing the sending of m_l ($\text{send}(p_j, m_l)$) and because of $\text{send}(p_j, m_n) \rightarrow \text{send}(p_j, m_l) \Rightarrow \text{delivery}(p_j, m_r) \rightarrow \text{send}(p_j, m_l)$, then $\text{bit_T}_k \notin \Phi(p_j)$ and therefore, $\text{bit_T}_k \notin h(m_l)$, which is a contradiction.

The following proposition shows that through the bits attached to the sent messages at wireless channels and the causal information stored at the base stations, we identify messages that satisfy the PC1 and PC2 propagation constraints with respect to m_l .

Proposition 1:

$$\text{bit_T}_k \in h(m_l) \Rightarrow (i, a) \in H(m_l)$$

Proof: By line 23, Table 4, we have that $(i, a, k') \in \text{CI}(\text{BS})$ only after the delivery of message $m_k = (i, a, \text{mes_received}(p_i), \text{event}, h(m_k))$ at the local base station BS. In this case, k' (by line 17, Table 4) identifies the sent message by the base station to its local mobile hosts. In the delivery of m_k at p_j with $a = k'$, we have (by Lines 1 and 11, Table 3) that $k' = \text{mes_received}(p_j)$ and by Line 12, Table 3, we have that $\text{bit_T}_k \in \Phi(p)$. We know by Lemma 2 (Step 2) that if $\text{bit_T}_k \in h(m_l)$, then $d(m_k, m_l) \leq \text{causal_distance}$. On the reception of message m_l sent by p_j with $m_l = (p_j, b, \text{mes_received}(p_j), \text{event}, h(m_l))$ at the base station BS, by Lines 8-13, Table 4, we have $(i, a) \in H(m_l)$ because there is in $\text{CI}(\text{BS})$ an element (i, a, k') where $k' = \text{mes_received}(p_j)$, such that (i, a) identifies to messages m_k with $d(m_k, m_l) \leq \text{causal_distance}$. \diamond

Lemma 3: For all $m_k, m_l \in M$, $m_k \rightarrow m_l$ such that $\text{Src}(m_k) \neq \text{Src}(m_l)$ and $\text{redundancy}(m_k) \leq \text{causal_distance}$ implies that $m_k = (i, a) \in H(m)$. This is accomplished by the procedures at the diffusion message by Lines 8, 11 and 18 and at the reception message by Lines 12 and 23, Table 4 and by lines 14 and 19, Table 5.

Theorem 2: (Correctness) for all $m_k, m_l \in M$, $m_k \rightarrow m_l$ such that $d(m_k, m_l) \leq \text{causal_distance}$ implies that $\text{delivery}(m_k) \rightarrow \text{delivery}(m_l)$.

Proof: Let us consider two messages m_0 and m_n such that $\text{send}(m_0) \rightarrow \text{send}(m_n)$ and both are received by p . We show that they are delivered to p according to causal ordering.

For this proof, we have two general cases. The proof is by induction on the distance $d(m_0, m_n)$.

Base case:

$$d(m_0, m_n) = 1 \text{ and } d(m_0, m_n) \leq \text{causal_distance}$$

In this case, m_0 is IDR related to m_n and from Lemma 3 and since always $d(m_k, m_l) \leq \text{redundancy}(m_k)$, we have $m_0 \in H(m_n)$. It follows that line 5, Table 5, will delay the delivery of m_n until after the delivery of m_0 .

Induction case:

$$d(m_0, m_n) \geq 2 \text{ and } d(m_0, m_n) \leq \text{causal_distance}$$

By induction, we have that all messages of the set $\{m_r \in M: m_{r-1} \downarrow m_r \text{ for all } r = 1 \dots n-1\}$ that are delivered to p are delivered in causal order. For the induction phase, we have two cases depending on whether m_{n-1} has been delivered or discarded at p :

- For m_{n-1} delivered at p . We have m_{n-1} that immediately precedes m_n so the base case applies to these messages: m_{n-1} is delivered before m_n and by transitivity m_0 is delivered before m_n
- For m_{n-1} discarded at p . In this case $m_{n-1} \in H(m_n)$ and by Lemma 3 and Line 5, Table 5, it follows that m_n is delivered after its lifetime has expired. We have that for a message m_r that belongs to the path m_0 to m_{n-1} implies that the delivery or discarded time of m_r is less than or equal to the discarded time of m_{n-1} , Line 5, Table 5. Consequently, m_n is delivered at p after m_0 .

We notice that when a message m_{n-y} such that $n-y > \text{causal_distance}$, we have $m_{n-y} \notin H(m_n)$ and therefore, we cannot ensure the causal delivery of m_{n-y} with respect to m_n .

DISCUSSION

Overhead analysis: In our protocol, the size of the control information sent over the wired communication channels depends on the number of concurrent messages that immediately precede a message m . In order to be efficient, each entry in $H(m)$ corresponds to the most recent message sent by a mobile host p_i and causally received by p_j (Lines 14-16, Table 4). This is possible since each message m is sequentially timestamped with its respective local logical clock of mobile host p_i (Line 3, Table 2). By knowing the sequential order, a mobile host p_j can determine at any message reception if a message or set of messages diffused by p_i has been lost, independently of the causal distance.

Since $H(m)$ has only the most recent messages that precede a message m , the overhead per message sent over the wired communication channels in our protocol to ensure causal ordering 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 exist), we note that the overhead per message is at most the causal distance established ($|H(m)| \leq \text{causal_distance}$) and for the case of concurrent messages, the worst case is at most $n-1$ ($|H(m)| \leq n-1$), which is the same boundary for messages that are IDR related ($\text{causal_distance} = 1$).

On the other hand, the control information attached to messages sent over the wireless network and stored at a mobile host can fluctuate between 0 and $n-1$ bits ($0 \leq h(m) \leq n-1$), where n is equal to the number of mobile hosts in the group. Again, the size of $h(m)$ depends on the number of concurrent messages that immediately precede a message. In the best case, which is the serial case, the size of $h(m)$ is equal to, at most, the causal distance established ($h(m) \leq \text{causal_distance}$) and in the worst case, for the case of concurrent messages, the size of $h(m)$ is at most $n-1$ bits.

We notice that in our Causal-FEC protocol, as for MOCIVI causal protocol in (Lopez *et al.*, 2008), the likelihood that the worst case will occur approaches zero as the number of participants in the group grows. This is because the likelihood that n concurrent messages occur decreases inversely proportional to the size of the communication group. This behavior has been shown in (Pomares *et al.*, 2004). Compared with protocols that are exclusively based on vector clocks (Mattern, 1988), our worst case denotes for them the constant overhead that must always be attached per message.

CONCLUSION

An efficient FEC causal protocol has been presented. The protocol is efficient in terms of the overhead attached to messages, computational cost and storage control information on the mobile hosts. In our study, the redundancy control information attached per message is dynamically adapted to the behavior of the system. We have shown that this control information allows us to perform a causal forward error recovery when messages are lost. Our Causal-FEC protocol is suitable for multimedia cooperative systems in real time since it performs a forward error recovery and it does not require previous knowledge of the behavior of the system.

REFERENCES

- Acharya, A. and B. Badrinath, 1996. A framework for delivering multicast messages in networks with mobile hosts. *ACM/Baltzer Mob. Networks Appli.*, 1: 199-219. DOI: 10.1007/BF01193337
- Alagar, S. and S. Venkatesan, 1997. Causal ordering in distributed mobile systems. *IEEE Trans. Comput.*, 46: 353-361. DOI: 10.1109/12.580430
- Alberto, B. and S. Francesco, 2001. A reliable multicast protocol for distributed mobile systems: design and evaluation. *IEEE Trans. Paral. Distrib. Syst.*, 12: 1009-1022. DOI:10.1109/71.963414
- Anastasi, G., A. Bartoli and L. Flaminia, 2004. Fault-tolerant support for reliable multicast in mobile wireless systems: Design and evaluation. *Wireless Networks*, 10: 259-269. DOI: 10.1023/B:WINE.0000023860.73046.94
- Baldoni, R., A. Mostefaoui and M. Raynal, 1996. Causal delivery of messages with real-time data in unreliable networks, real-time systems. *Int. J. Time-Crit. Comput. Sys.*, 10: 245-262. DOI: 10.1007/BF00383387
- Benzaid, C. and N. Badache, 2005. Mobi-causal: A protocol for causal message ordering in mobile computing systems. *Mobile Comput. Commun. Rev.*, 9: 19-28. DOI: 10.1145/1072989.1072993
- Chandra, P. and A.D. Kshemkalyani, 2004. Causal multicast in mobile networks. *Proceeding of the IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications System*, Oct. 213-220, IEEE Computer Society, Washington DC., USA., pp: 4-8. DOI: 10.1109/MASCOT.2004.1348235
- Ghim, Y., T. Ah-Hwee and P. Hwee-Hwa, 2007. Discovering and exploiting causal dependencies for robust mobile context-aware recommenders. *IEEE Trans. Knowl. Data Eng.*, 19: 977-992. DOI: 10.1109/TKDE.2007.1065
- Kuang, C., Y. Li, T. Chao and H. Lu, 2000. A causal multicast protocol for mobile distributed systems. *IEICE Trans. Inform. Syst.*, 83: 2065-2073.
- Lamport, L., 1978. Time, clocks and the ordering of events in a distributed system. *Commun., ACM*, 21: 558-565. DOI: DOI: 10.1145/359545.359563
- Lopez, E., J. Estudillo, J. Fanchon and S.E. Pomares, 2005. A fault-tolerant causal broadcast algorithm to be applied to unreliable networks. *Proceeding of the 17th International Conference on Parallel and Distributed Computing and Systems*, Nov. 14-16, Acta Press Publisher, Phoenix, AZ., USA., pp: 465-470. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.77.4368>
- Lopez, E., S. Pomares and G. Rodriguez, 2008. MOCIVI: An efficient causal protocol to cellular networks. *Int. J. Comput. Sci. Network Secur.* 8: 136-144. http://paper.ijcsns.org/07_book/200801/20080120.pdf
- Mattern, F., 1988. Virtual time and global states of distributed systems. *Proceeding of the Workshop on Parallel and Distributed Algorithms*, Elsevier, Chateau De Bonas, France, pp: 215-226. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.7435>

- Perkins, C., 2003. RTP Audio and Video for Internet. 1st Edn., Addison Wesley, USA., ISBN: 0-672-32249-8, pp: 432.
- Pomares, S., E. Lopez, G. Rodriguez and J. Fanchon, 2009. A efficient Δ -causal algorithm for real time distributed system. *J. Applied Sci.*, 9: 1711-1718. DOI: 10.3923/jas.2009.1711.1718
- Pomares, S., J. Fanchon and K. Drira, 2004. The Immediate dependency relation: An optimal way to ensure causal group communication. *Ann. Rev. Scalable Comput. Edn. World Scient. Seri. Scalable Comput.*, 6: 61-79. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.3209>
- Prakash, R., M. Raynal and M. Singhal, 1996. An efficient causal ordering algorithm for mobile computing environments. Proceedings of the 16th International Conference on Distributed Computing Systems, IEEE Computer Society, Washington DC., USA., pp: 744-751. DOI: 10.1109/ICDCS.1996.508027
- Tarkoma, S., 2003. Distributed event dissemination for ubiquitous agents. Proceedings of the 10th ISPE International Conference on Concurrent Engineering, A.A. Balkema Publisher, Madeira, Portugal, pp: 105-110. DOI: 10.1.1.98.4603
- Yi, C., S. Chiand and S. Yen, 2003. An efficient time-based checkpointing protocol for mobile computing systems over mobile IP. *Mobile Networks Appli.*, 8: 687-697. DOI: 10.1023/A:1026086712672