

# An Efficient Two-Tier Causal Protocol for Mobile Distributed Systems

Eduardo Lopez Dominguez<sup>1\*</sup>, Saul E. Pomares Hernandez<sup>2,3,4</sup>, Gustavo Rodriguez Gomez<sup>2</sup>, Maria Auxilio Medina<sup>5</sup>

**1** Research and Innovation Centre, Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, Mexico, **2** Computer Science Department, Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, Mexico, **3** LAAS-CNRS, Toulouse, France, **4** Univ. de Toulouse, Toulouse, France, **5** Computer Science Department, Universidad Politécnica de Puebla, San Mateo, Puebla, Mexico

## Abstract

Causal ordering is a useful tool for mobile distributed systems (MDS) to reduce the non-determinism induced by three main aspects: host mobility, asynchronous execution, and unpredictable communication delays. Several causal protocols for MDS exist. Most of them, in order to reduce the overhead and the computational cost over wireless channels and mobile hosts (MH), ensure causal ordering at and according to the causal view of the Base Stations. Nevertheless, these protocols introduce certain disadvantage, such as unnecessary inhibition at the delivery of messages. In this paper, we present an efficient causal protocol for groupware that satisfies the MDS's constraints, avoiding unnecessary inhibitions and ensuring the causal delivery based on the view of the MHs. One interesting aspect of our protocol is that it dynamically adapts the causal information attached to each message based on the number of messages with immediate dependency relation, and this is not directly proportional to the number of MHs.

**Citation:** Dominguez EL, Hernandez SEP, Gomez GR, Medina MA (2013) An Efficient Two-Tier Causal Protocol for Mobile Distributed Systems. PLoS ONE 8(4): e59904. doi:10.1371/journal.pone.0059904

**Editor:** Rodrigo Huerta-Quintanilla, Cinvestav-Merida, Mexico

**Received:** December 26, 2012; **Accepted:** February 20, 2013; **Published:** April 9, 2013

**Copyright:** © 2013 Lopez Dominguez et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** The authors have no support or funding to report.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: elopez@lania.mx

## Introduction

The deployment of mobile distributed systems (MDS), in conjunction with wireless communication technologies and Internet, enables portable computing devices (referred in this paper as *mobile hosts*), such as smart phones and personal digital assistants (PDAs), to communicate from anywhere and at anytime. The MDS 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, 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 three main aspects: host mobility, asynchronous execution, and unpredictable communication delays. Causal ordering 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 in several domains, such as ubiquitous agents systems [1], context-aware systems [2] and multimedia synchronization protocols [3].

In this paper, we consider the problem of causal ordering message delivery among mobile hosts in the context of group communication. Recently, some protocols have been proposed to implement causal message ordering for mobile distributed systems [4], [5], [6], [7], [8], [9]. Nevertheless, most of these protocols, in order to reduce computational cost and communication loads on

mobile hosts, ensure causal ordering at and according to the Base Stations (BSs). These methods give rise to two main problems. First, the causal order seen by the MHs (referenced in this paper as *mobile causal view*) greatly differs from the causal orders of messages in which they were originally sent. Secondly, these methods introduce unnecessary inhibition at the message delivery.

Other important aspect concerning the design of causal protocols for a MDS is mobility management. When a mobile host moves from a cell (source) to another cell (target), these protocols must continue to ensure the causal order of messages. To achieve this, they execute a handoff module. This module mainly consists on sending the main structures used between the source and target cells. However, most of them stop the sending of new messages among all mobile hosts during the execution of this procedure.

In this work, we propose a new protocol that ensures the causal ordering according to the causal view that the mobile hosts perceive in the MDS, avoiding the unnecessary inhibition at the message delivery, while maintaining a low overhead and computational cost. To achieve this, our causal protocol works at two communication levels according to the connection type: intra-base communication level and inter-base communication level. At the intra-base communication level (wireless connection), we only send as causal overhead timestamped per message, between a BS and the MHs, a structure of bits denoted by  $h(m)$ . The  $h(m)$  is dynamically determined based on the immediate dependency relation, IDR [10]. In the best case, the size of  $h(m)$  is equal to 1 bit; and in the worst case, it is equal to  $n$  bits, ( $1 \leq |h(m)| \leq n$ ), where  $n$  is the number of *MHs* in the system.

At the inter-base communication level (wired connection) the causal overhead  $H(m)$  sent per message between BSs is composed of entries of the form  $(p, t)$ , which are message identifiers, where  $p$  is the mobile host identifier, and  $t$  is the logical local clock of mobile host  $p$ . The size of the causal overhead  $H(m)$  is also dynamically determined by the IDR ( $1 \leq |H(m)| \leq n$ ). On the other hand, we propose a handoff management process that is characterized by allowing an asynchronous transfer of the mobile hosts among the cells of the system. This handoff management process does not interrupt the communication at any moment.

This paper is organized as follows. Section 2 presents the preliminaries (system model, background and definitions). The mobile causal protocol is provided in Section 3. In Section 4, we compare our protocol with the related works. Finally, conclusions are presented in Section 5.

## Materials and Methods

### The System Model

We consider that a MDS runs over a wireless network infrastructure, which consists of two kinds of entities: base stations (BSs) and mobile hosts (MHs). A BS communicates with mobile hosts through wireless communication channels. The geographic area covered by a BS is called a cell, and it is depicted in Figure 1. At any given time, a MH is assumed to be within the cell of at most one BS, which is called its local BS. A MH can communicate with other MHs and BSs only through its local BS.

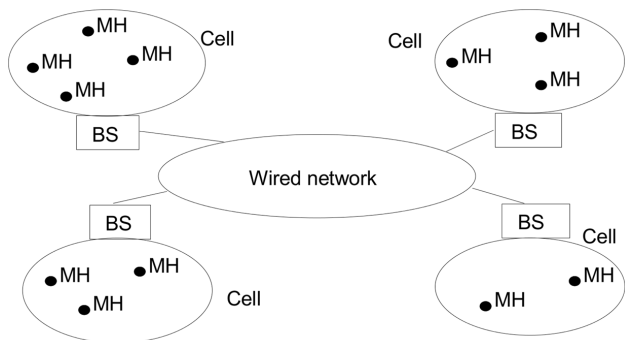
The base stations are connected among themselves using wired channels. The BSs and the wired channels constitute the static network. We assume that the wired channels are reliable, with an arbitrary but finite amount of time to deliver messages. Due to system asynchrony and unpredictable communication delays, the messages on a MDS from MH to MH can arrive in a different order as they were sent. In a mobile distributed system, a mobile host can move from one BS to another. In this case, a handoff procedure is performed to transfer the communication responsibilities of a MH to the new BS.

### Background and Definitions

Causal ordering delivery is based on the happened-before relation (HBR) defined by Lamport [11]. This relation establishes causal precedence dependencies over a set of events 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  $a \rightarrow b$ .



**Figure 1. Physical architecture of a MDS.**  
doi:10.1371/journal.pone.0059904.g001

- 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 is 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  $send(m) \rightarrow send(m')$ .

**The Immediate Dependency Relation.** The Immediate Dependency Relation (IDR) [10] 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):

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

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  $m'$ .

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

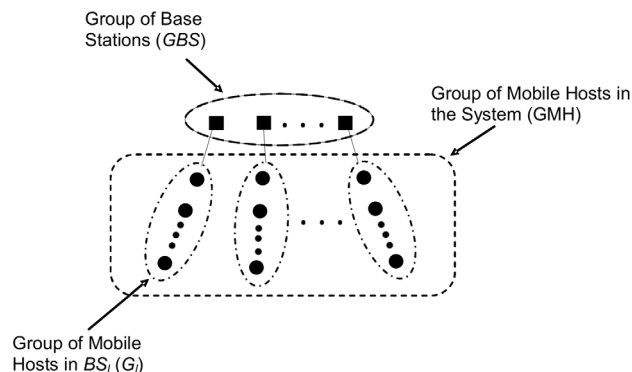
$$\begin{aligned} \text{if } \forall m, m' \in M, m \downarrow m' \Rightarrow \forall p \in g : \text{delivery}(p, m) \rightarrow \text{delivery}(p, m') \text{ then} \\ \forall m, m' \in M, m \rightarrow m' \Rightarrow \forall p \in g : \text{delivery}(p, m) \rightarrow \text{delivery}(p, m') \end{aligned}$$

## Results

### Protocol composition

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. The  $GMH$  is subdivided into subgroups ( $G_i$ ); one for each  $BS$  (see Figure 2).

The  $BS$ s in the  $GBS$  and the mobile hosts in a  $G_i$  communicate by reliable 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



**Figure 2. Logical structure considered for a mobile distributed system.**  
doi:10.1371/journal.pone.0059904.g002

specific process with its respective identifier, we write  $p_i$ . The set of destinations of a message  $m$  is always a *GMH*.

In our work, the *GBS* carries out the causal delivery of messages according to the order in which messages were observed by the mobile hosts. To achieve this, each mobile host  $p$  uses a structure of bits  $\Phi(p)$  in order to establish an immediate dependency relation (Definition 2) among messages. The content of  $\Phi(p)$  is the only control information attached per message in the wireless channel ( $1 \leq |\Phi(p)| \leq n$ ). Each bit in  $\Phi(p)$  identifies a causal message  $m$  that has a potential IDR with the next message to be sent by  $p$ . The base stations keep the main structures of causal control information of the mobile distributed system, such as the vector clock *VT* introduced by Mattern [12]. Through the control information and the structure of bits  $h(m) \leftarrow \Phi(p)$  sent in the system, the base stations can determine the immediate dependency relation among the messages sent by *MHS* on different *BSs*.

### Data Structures

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

- $mes\_received(p)$  is a counter, which is incremented 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. Each bit in  $\Phi(p)$  identifies a message  $m$  in the causal past of  $p$  that has a potential *IDR* with the next message to be sent by  $p$ . The size of  $\Phi(p)$  fluctuates between  $1 \leq |\Phi(p)| \leq n$ .

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 clock of Mattern [12]. 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 a control information structure. It is a set of entries  $(i, t, d, ip)$ . The entry  $(i, t, d, ip)$  represents a message sent by the mobile host  $p_i$  with logical local clock  $t = VT(BS)[i]$  and  $d = mes\_sent(BS)$ . Finally,  $ip$  is a Boolean variable. The  $BS$  sets  $ip$  to *true* when it detects that a recently message received has an immediately dependency relation with the message  $m = (i, t, d, ip)$ . In Section protocol description we present a detailed description of how this procedure is carried out.

**Message structures.** The following message structures are used in the *MDS* by the mobile hosts and base stations (see Figure 3).

- The messages sent in the wireless communication channels by mobile hosts to their 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. Each bit in  $h(m)$  identifies a message in the causal past of  $p_i$  that has IDR with  $m$ .
- A message  $m$  sent among base stations *BSs* is denoted by  $bs(m)$ , and it is composed by a quadruplets  $bs(m) \equiv (i, t, data, H(m))$ , where the structures  $i, t$  have been previously described and:
  - $data$  is the content of the message, and

- $H(m)$  is composed of a set of elements  $(i, t)$ , which represent messages that have an IDR with  $m$ .
- A message  $m$  received by a  $BS_i$  from a mobile host  $p \in G_i$  and which has been resent by such  $BS_i$  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. Each bit in  $h^*(m)$  identifies a message in the causal past of  $p_i$  that has an IDR with  $m$  and that the  $BS_i$  has not ensured its causal delivery.
- A message  $bs(m)$  received by a  $BS_i$  and which has been resent within its cell, consists of a quintuplet that we call  $inter(m) \equiv (i, t, data, h^*(m))$ .

### Specification of the MOKA Protocol

Structures and variables at mobile host  $p_i$  are initialized as follows:

- $\Phi(p_i) \leftarrow \emptyset$
- $bit \leftarrow 1b$
- $mes\_received(p_i) = 0$
- $mes\_sent(p_i) = 0$

Structures and variables at  $BS_r$  are initialized as follows:

- $VT(BS_r)[i] = 0 \forall i : 1 \dots n;$
- $CI(BS_r) \leftarrow \emptyset$
- $H(m) \leftarrow \emptyset$
- $mes\_sent(BS_r) = 0$

Next, we present in Tables 1–4 our causal protocol for groupware which satisfies the *MDS*'s constraints, avoiding unnecessary inhibitions and ensuring the causal delivery based on the view of the *MHS*.

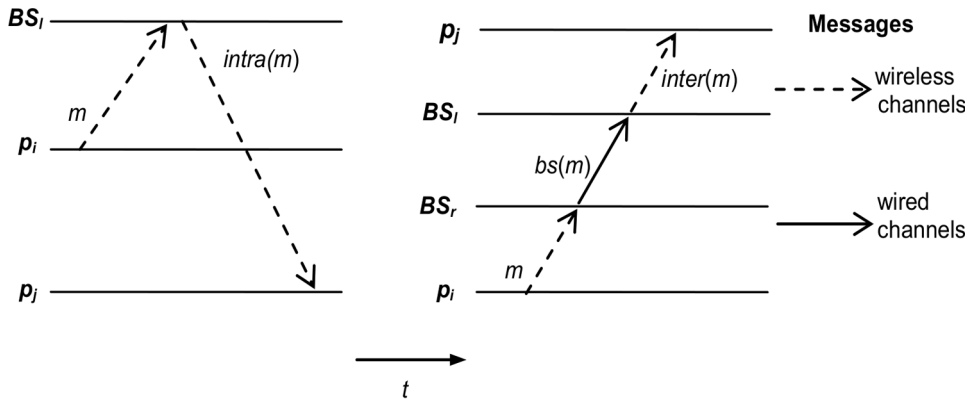
### Protocol Description

The main contribution of the present paper is to ensure causal ordering according to the causal view at the *GMH*. In this section, we focus on how our protocol performs that causal ordering at the wireless network level. A description of the causal ordering algorithm for the wired network i.e. the group of base stations level is presented in [10].

In this Section we will refer to Figure 4 in order to explain how our protocol ensures causal ordering. In this scenario, the group of mobile hosts is composed by  $GMH = \{p_1, p_2, p_3, p_4\}$  and the group of base stations is integrated by  $GBS = \{BS_1, BS_2\}$  where  $p_1, p_2 \in BS_1$  and  $p_3, p_4 \in BS_2$ . In order to show how our protocol ensures the causal order, we focus on the message  $m_5$  sent by  $p_3$  and its delivery to the mobile hosts at  $BS_1$ .

Prior to the delivery of  $m_5$  to  $BS_2$ , the control information at  $BS_1$  and  $BS_2$  are:  $CI(BS_1) = (\{p_1, 1, 2, true\}, \{p_2, 1, 3, false\})$ ,  $CI(BS_2) = (\{p_1, 1, 2, true\}, \{p_4, 1, 3, false\}, \{p_2, 1, 4, false\})$ ,  $VT(BS_1) = (1, 1, 1, 0)$  and  $VT(BS_2) = (1, 1, 1, 1)$ . These values are deduced from our protocol shown in Tables 1–4; see Section specification of the Moka protocol. According to our algorithm, the delivery process is described as follows.

At the diffusion of message  $m_5$  by  $p_3$  to  $BS_2$ , **Table 1, Lines 1–6** (henceforth, we will use “Lines 1, 1–6” to simplify), the value of  $mes\_sent(p_i)$  is increased by one,  $mes\_sent(p_i) = 2$ , (**Line 1, 1**). The mobile host  $p_3$  copies the bits structure  $\Phi(p_3)$  to  $h(m_5)$ , (**Line 1, 2**). Message  $m_5 = (p_3, 2, 4, data, h(m_5) = 11)$  is constructed and sent in



**Figure 3. Messages sent at the intra-base and inter-base levels, respectively.**  
doi:10.1371/journal.pone.0059904.g003

(Line 1, 3–4). Through  $h(m_5) = 11$  local  $BS_2$  will be able to determine which messages immediately precede  $m_5$ .

When message  $m_5$  is received at  $BS_2$ , the FIFO delivery condition is verified (Line 3, 2). From our scenario, this condition is satisfied. Then the message  $m_5$  is delivered to  $BS_2$  and the  $VT(BS_2)$  vector is increased by one at the position  $p_3$ ,  $VT(BS_2) = (1, 1, 2, 1)$ . Later on, the  $BS_2$  sends message  $m_5$  to  $BS_1$ . This is done through the diffusion of message  $bs(m_5)$  by  $BS_2$ .

The message  $bs(m_5)$  is constructed by  $BS_2$  as follows. According to  $h(m_5)$  structure, there are two messages that immediately precede  $m_5$ . In order to identify these messages, (Lines 3, 7–15),  $BS_2$  determines if there are some elements in  $CI(BS_2)$  with  $d$  equal to variable  $mes\_received(p_3)$  of  $m_5$ . In this case, there is an element at  $CI(BS_2)$  related to  $p_2$  with  $d = 4$  (Line 3, 11). Afterwards, the variable  $mes\_received(p_3)$  is decremented by one (Line 3, 13). In the next iteration of the algorithm,  $BS_2$  found another element at  $CI(BS_2)$  with respect to  $p_4$  with  $d = 3$  (Line 3, 11). Therefore, the only control information attached to  $bs(m_5)$  in order to ensure a causal order relates to  $m_3$  and  $m_4$ , which are the only messages that have an immediate dependency relation with  $bs(m_5)$ , see Figure 4. Hence, the message sent from  $BS_2$  to  $BS_1$  is  $bs(m_5) = (p_3, 2, data, H(m_5) = (\{p_2, 1\}, \{p_4, 1\}))$  (Line 3, 17).

When message  $bs(m_5)$  is received at base station  $BS_1$ , see Figure 4,  $BS_1$  verifies that the message satisfies the FIFO and causal delivery condition, (Line 4, 2). In this case, message  $bs(m_5)$  satisfies only the FIFO delivery condition (Line 4, 2) because  $t = 2$  and  $VT(BS_1)[p_3] = 1$ , ( $t = VT(BS_1)[p_3] + 1$ ). Due to the fact that message  $m_4$  has not been received by mobile hosts within the cell of  $BS_1$ , the causal delivery condition ( $1 \leq VT(BS_1)[p_4] = 0$ ) is not satisfied; therefore the message  $bs(m_5)$  cannot be delivered causally and it is delayed (Line 4, 3).

According to this scenario, message  $bs(m_4)$  is received by  $BS_1$ , (Lines 4, 1–14).  $BS_1$  verifies that message  $bs(m_4)$  satisfies the FIFO and causal delivery condition. In this case,  $bs(m_4) = (p_4, 1, BS_2, data, H(m_4) = (p_1, 1))$  satisfies both conditions (Lines 4, 2). Therefore, message  $bs(m_4)$  is delivered, and the vector is increased by one in  $VT(BS_1)[p_4]$  resulting in  $VT(BS_1) = (1, 1, 1, 1)$ . Later on,  $BS_1$  sends the message  $bs(m_4)$  to its local mobile hosts. The message to be sent by  $BS_1$  to local mobile hosts is  $inter(m_4) = (p_4, 4, data, h'(m_4) = 0)$ , (Lines 4, 12).

The delivery of message  $inter(m_4)$  by mobile host  $p_1$  is as follows, (Lines 2, 1–8). The mobile host  $p_1$  updates  $\Phi(p_1)$  with the attached information of message  $inter(m_4)$  (Lines 2, 6–8). The structure of bits after updating the data structures at  $p_1$  is  $\Phi(p_1) = 11$ . Afterwards, the variable  $mes\_received(p_1)$  is increased by one,  $mes\_received(p_1) = 4$ .

Finally, after the delivery of message  $bs(m_4)$  to mobile host  $p_1$ ,  $BS_1$  verifies if message  $bs(m_5)$  satisfies the causal delivery condition (Line 4, 2). Now, message  $bs(m_5)$  satisfies the causal delivery condition,  $1 \leq VT(BS_1)[p_4] = 1$ , because of message  $m_4$  has been received by mobile hosts within the cell covered by  $BS_1$ . Therefore, the message  $bs(m_5)$  can be delivered causally.  $BS_1$  must send the message  $bs(m_5)$  to its local mobile hosts. The message sent by  $BS_1$  to local mobile hosts is  $inter(m_5) = (p_3, 5, data, 11)$ , (Line 4, 12). When message  $inter(m_5)$  is received by mobile host  $p_1$ , (Lines 2, 1–8), its delivery is done in the same way as it was previously described for  $inter(m_4)$ .

**Table 1. Diffusion of message  $m$  by a mobile host  $p_i$ .**

<b>For each diffusion of message <math>send(m)</math> at mobile host <math>p_i</math></b>	
1.	$mes\_sent(p_i) = mes\_sent(p_i) + 1$
2.	$h(m) \leftarrow \Phi(p_i)$
3.	$m = (i, t = mes\_sent(p_i), mes\_received(p_i), data, h(m))$
4.	<b>Diffusion:</b> $send(m)$ /* sent of message $m$ to local $BS^*$ /
5.	$\Phi(p_i) \leftarrow \emptyset$
6.	$mes\_received(p_i) = mes\_received(p_i) + 1$

doi:10.1371/journal.pone.0059904.t001

**Table 2. Reception of message  $intra(m)$  or  $inter(m)$  by a mobile host  $p_j$ ,  $i \neq j$ .**

/* $intra(m) = (i, t, data, h'(m))$ or $inter(m) = (i, t, data, h'(m))$ */	
1.	<b>if</b> not ( $t = mes\_received(p_j) + 1$ ) then
2.	wait ( $intra(m) \mid inter(m)$ )
3.	<b>Else</b>
4.	$delivery(intra(m) \mid inter(m))$
5.	$mes\_received(p_j) = mes\_received(p_j) + 1$
6.	$\mu \{bit\} \in h'(m)$
7.	$\Phi(p_j) \leftarrow \Phi(p_j) / \{bit\}$
8.	$\Phi(p_j) \leftarrow \Phi(p_j) \cup \{bit\}$ <b>endif</b>

doi:10.1371/journal.pone.0059904.t002

**Table 3.** Reception of message  $m = (i, t, mes\_received(p_i), data, h(m))$  and sending of  $intra(m)$  by a base station  $BS_r$ .

1.	<b>if</b> $i \in BS_r$ , then
2.	<b>if</b> not $(t = VT(BS_r)[i] + 1)$ then
3.	$wait(m)$
4.	<b>Else</b>
5.	$delivery(m)$
6.	$VT(BS_r)[i] = VT(BS_r)[i] + 1$
7.	$\mu \{bit\} \in h(m)$
8.	<b>if</b> $\exists (k, t, d, ip) \in C(BS_r) \mid d = mes\_received(p_i)$ and $ip = false$ then
9.	$h'(m) \leftarrow h'(m) \cup \{bit\}$
10.	$(k, t, d, ip) \leftarrow (k, t, d, ip = true)$ <b>endif</b>
11.	<b>if</b> $\exists (k, t, d, ip) \in C(BS_r) \mid d = mes\_received(p_i)$ then
12.	$H(m) \leftarrow H(m) \cup (k, t)$ <b>endif</b>
13.	$mes\_received(p_i) = mes\_received(p_i) - 1$
14.	$intra(m) = (i, t = mes\_sent(BS_r) + 1, data, h'(m))$
15.	$bs(m) = (i, t, data, H(m))$
16.	<b>Diffusion:</b> $send(intra(m))$ /* sending of message $intra(m)$ to local mobile hosts */
17.	<b>Diffusion:</b> $send(bs(m))$ /* sending of message $bs(m)$ to the other bases stations */
18.	<b>endif</b>
19.	$mes\_sent(BS_r) = mes\_sent(BS_r) + 1$
20.	$C(BS_r) \leftarrow C(BS_r) \cup \{ (i, t, mes\_sent(BS_r), ip = false) \}$ <b>endif</b>

doi:10.1371/journal.pone.0059904.t003

### Handoff Management Process

When a mobile host  $p$  in a cell covered by  $BS_r$  moves to a cell covered by  $BS_s$ , the responsibility of maintaining its causal dependencies shifts from the base station  $BS_r$  to  $BS_s$ . In order to ensure a causal ordering of messages in a mobile distributed system, the handoff module described in this Section is executed. In our case, we send only a control message with causal information about  $p$  in order to ensure a causal ordering of

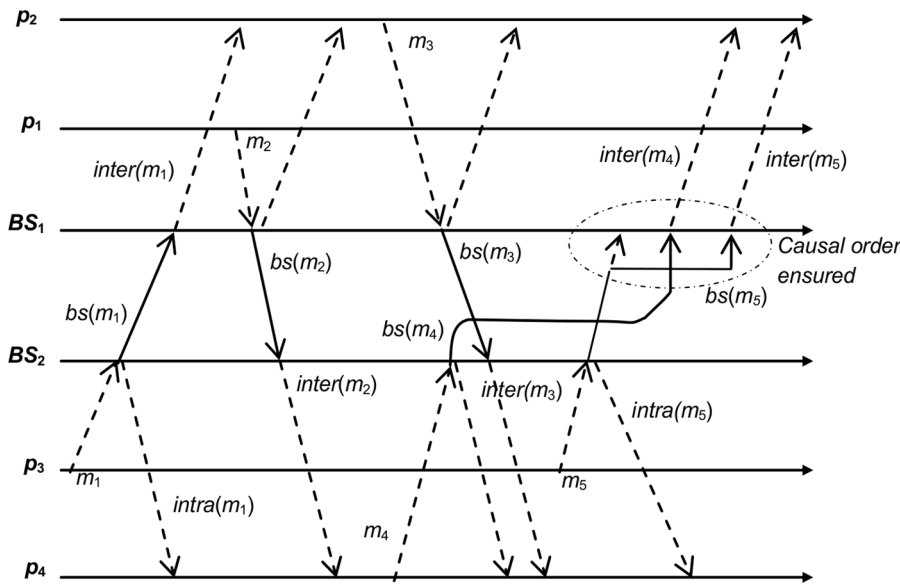
messages at the group of mobile hosts ( $GMH$ ). The steps carried out by the handoff management process are depicted in Figure 5.

Assume that a mobile host  $p$  located in the cell covered by the source base station  $BS_r$  moves to a cell covered by the base station  $BS_s$  (see Figure 5). The first step established by  $p$  is to send the message  $handoff\_begin = (p, t, BS_r, h(handoff\_begin) = \Phi(p))$  to its  $BS_r$  (Figure 5). Upon receiving this message,  $BS_r$  informs  $BS_s$  and other base stations in the  $GBS$  that  $p$  is switching from base station  $BS_r$  to  $BS_s$  by sending the message  $handoff\_transfer = (p, t, H(handoff\_begin))$ ,

**Table 4.** Reception of message  $bs(m) \equiv (i, t, data, H(m))$  and sending of  $inter(m)$  by a base station  $BS_r$ , such that  $i \notin BS_r$ .

1.	<b>if</b> $i \notin BS_r$ , then
2.	<b>if</b> not $((t = VT(BS_r)[i] + 1)$ and $(\mu (s, x) \in H(m): x \leq VT(BS_r)[s]))$ then
3.	$wait(bs(m))$
4.	<b>else</b>
5.	$delivery(bs(m))$
6.	$VT(BS_r)[i] = VT(BS_r)[i] + 1$
7.	$\mu (x, y) \in H(m)$
8.	<b>if</b> $\exists (k, t, d, ip) \in C(BS_r) \mid x = k$ and $y = t$ and $ip = false$ then
9.	$h'(m) \leftarrow h'(m) \cup \{bit\}$
10.	$(k, t, d, ip) \leftarrow (k, t, d, ip = true)$ <b>endif</b>
11.	$inter(m) = (i, t = mes\_sent(BS_r) + 1, data, h'(m))$
12.	<b>Diffusion:</b> $send(inter(m))$ /* sending of message $inter(m)$ to local mobile devices */
13.	<b>endif</b>
14.	<b>endif</b>
15.	$mes\_sent(BS_r) = mes\_sent(BS_r) + 1$
16.	$C(BS_r) \leftarrow C(BS_r) \cup \{ (i, t, mes\_sent(BS_r), ip = false) \}$

doi:10.1371/journal.pone.0059904.t004



**Figure 4. Scenario of communication group composed by four mobile hosts and two BSs.**  
doi:10.1371/journal.pone.0059904.g004

where  $H(handoff\_begin)$  is the causal history of mobile host  $p$ , see Figure 5. The structure  $H(handoff\_begin)$  contains the identifiers of the last messages received by  $p$ , when it was over the cell covered by  $BS_s$ .

In order to ensure the causal order, when the target  $BS_t$  receives the message  $handoff\_transfer = (p, t, H(handoff\_begin))$ , it verifies that this message satisfies the causal delivery condition. If the message satisfies the causal delivery condition, then it is delivered and  $BS_t$  makes the registration of  $p$  as a new mobile host in its cell, see Figure 5. Otherwise, the message is delayed until the causal condition is satisfied. The ending of the handoff procedure is identified by the diffusion of message  $handoff\_end(p)$  by  $BS_s$  to the other base stations, see Figure 5. After this message,  $BS_s$  takes care of maintaining the causal ordering of messages sent to  $p$ .

In our handoff management process, we attach causal information about  $p$  to message  $handoff\_transfer$  sent by the  $BS_s$ . This causal information is used by the target base station ( $BS_t$ ) in order to determine if the mobile host served by  $BS_t$  has received the messages observed by the mobile host  $p$  when it was over the cell covered by the source base station ( $BS_s$ ). If the message  $handoff\_transfer$  satisfies the causal delivery condition, the mobile host  $p$  can be registered by the target base station because the messages observed by  $p$  have been received by the mobile hosts served by  $BS_s$ . Otherwise, the mobile host  $p$  cannot be registered by the target base station because the messages observed by mobile host  $p$  when it was in the source base station can be received again by it, violating the causal order.

In our case, we do not attach data structures, such as vector clock ( $VT(BS)$ ), to the messages sent during the handoff management process in order to maintain the causal ordering. Instead, we attach to the messages the causal information that includes the messages with immediate dependency relation. This allows us to continue with the system execution without waiting for the handoff procedure to end. Therefore, the handoff management process that we propose is characterized by allowing an asynchronous transfer of the mobile hosts among the cells of the system. Moreover, this handoff management process does not interrupt the communication at any moment. Therefore, our handoff management process is asynchronous.

**Correctness Proof**

To show that our algorithm ensures the causal delivery (correctness), we provide a correctness proof. In order to do the proof as simple as possible, we focus on the novel part for the wireless channels, which is the information (bits) attached to the messages and the causal information stored at the base stations. We show that with this information we ensure the causal order.

Let two messages  $m_k = (p_i, a, event, h(m_k))$  and  $m_l = (p_j, b, event, h(m_l))$ , 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.**

$\forall bit_k \in h(m_l) \exists (x, y, k') \in CI(BS)$  such that  $k = k'$ , where  $(x, y)$  identifies a message  $m_k$ , which has IDR with  $m_l$ .

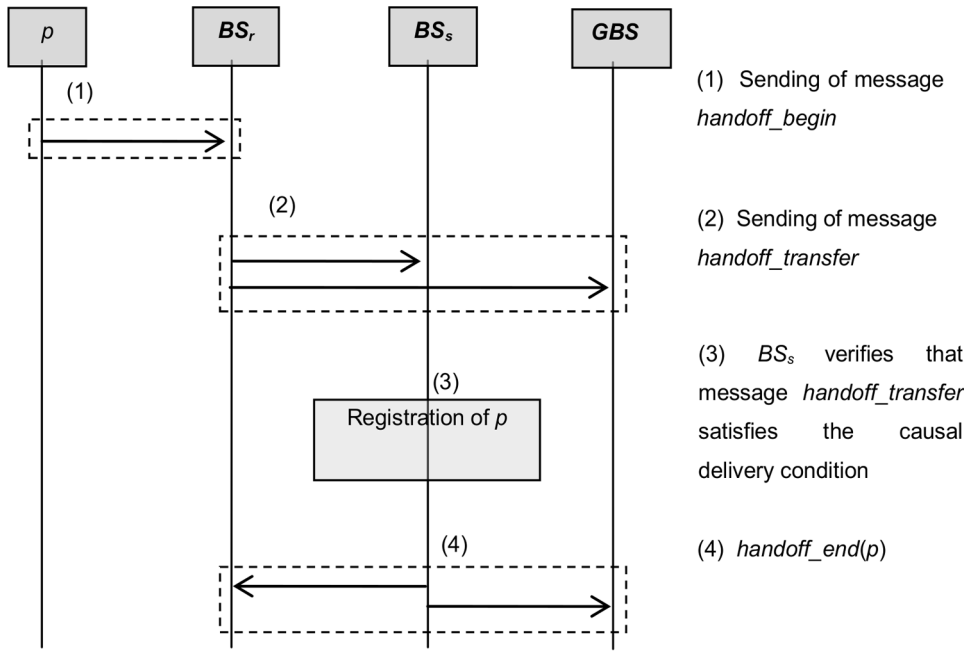
**Main steps of the proof.** The proof is composed by two lemmas and a proposition. The lemmas are intermediate results necessary for our proof:

- Lemma 1 shows that if  $bit_k$  belongs to the causal history of a message  $m_l$ , then the message identified by  $bit_k$  causally precedes message  $m_l$ .
- Lemma 2 indicates that the message  $m_k$  has an immediate dependency relation with the other message  $m_l$  if and only if the  $bit_k$  belongs to the causal history of the message  $m_l$ .
- Proposition 1 shows that through the bits structure ( $h(m)$ ) attached to sent messages and the causal information at the base stations, we ensure the causal order (theorem 1).

**Lemma 1.**

$$bit_k \in h(m_l) \Rightarrow m_k \rightarrow m_l$$

Proof: By Line 1, 2, we have that  $bit_k \in h(m_l)$  if and only if  $bit_k \in \Phi(p_j)$  when send of  $m_l$  is carried out by  $p_j$ , we denote it by  $send(p_j, m_l)$ . By using Line 2, 8, we have that  $bit_k \in \Phi(p_j)$  only after



**Figure 5. Description of the handoff module.**  
doi:10.1371/journal.pone.0059904.g005

the delivery  $m_k = (p_i, a, event, h(m_k))$  at  $p_j$ . This implies that the delivery of  $m_k$  precedes the sending of  $m_l$  ( $delivery(p_j, m_k) \rightarrow send(p_j, m_l)$ ). Therefore,  $m_k \rightarrow m_l$ .

**Lemma 2.**

$$m_k \downarrow m_l \Leftrightarrow bit_k \in h(m_l)$$

$$bit_k \in h(m_l) \Rightarrow m_k \downarrow m_l$$

The proof to this lemma is divided into two steps: First, we show that  $m_k \downarrow m_l \Rightarrow bit_k \in h(m_l)$  and second, we show that  $bit_k \in h(m_l) \Rightarrow m_k \downarrow m_l$ .

Step 1:

$$m_k \downarrow m_l \Rightarrow bit_k \in h(m_l)$$

The proof is by contrapositive, we proof that  $bit_k \notin h(m_l) \Rightarrow \exists m_r$  such that  $m_k \rightarrow m_r \rightarrow m_l$ ; thus, the message  $m_k$  has not an immediate independency relation with message  $m_l$ , see Section background and definitions. We assume that  $bit_k \notin h(m_l)$ . Only two events can delete  $bit_k$  of  $\Phi(p_j)$  before sending  $m_l$  ( $send(p_j, m_l)$ ), these are:

- By Lines 2, 6–7,  $bit_k$  is removed from  $\Phi(p_j)$  when the delivery of message  $m_r$  is carried out with  $bit_k \in h(m_r)$  at  $p_j$  ( $delivery(p_j, m_r)$ ). Lemma 1 shows that in this case  $m_k \rightarrow m_r$ . Moreover,  $delivery(p_j, m_r) \rightarrow send(p_j, m_l)$  implies that  $m_r \rightarrow m_l$  and then  $m_k \rightarrow m_r \rightarrow m_l$ . Therefore,  $m_k$  does not directly precede message  $m_l$ .
- By Line 1, 5, the sending of  $m_r$  at  $p_j$  empty  $\Phi(p_j)$ . In addition, the event of  $send(p_j, m_r)$  takes place such that  $delivery(p_j, m_k) \rightarrow send(p_j, m_r) \rightarrow send(p_j, m_l)$ . Therefore,  $m_k$  does not directly precede message  $m_l$ .

If neither of these two events occur, we have that  $bit_k \in \Phi(p_j)$  when the  $send(p_j, m_l)$  is carried out and by Line 1, 2, we have that  $bit_k \in h(m_l)$ .

Step 2:

The proof is by contradiction. By lemma 1, we know that if  $bit_k \in h(m_l)$  then  $m_k \rightarrow m_l$  with  $p_i \neq p_j$ . We suppose that there is a message  $m_r$  such that  $send(p_i, m_k) \rightarrow send(p_r, m_r) \rightarrow send(p_j, m_l)$ , and in addition that  $m_k \downarrow m_r$ . The proof considers two cases:  $p_r \neq p_j$  and  $p_r = p_j$ .

- We consider the case where  $p_r \neq p_j$  and the delivery  $m_k$  causally precedes to  $m_r$  ( $delivery(p_j, m_k) \rightarrow delivery(p_j, m_r)$ ) at  $p_j$ . By the step 1, we know that  $bit_k \in h(m_r)$ . Hence, on the delivery  $m_r$  ( $delivery(p_j, m_r)$ ) at mobile host  $p_j$ ,  $bit_k$  is deleted by Lines 2, 6–7. When performing the sending of  $m_l$  ( $send(p_j, m_l)$ ) and because of  $send(p_r, m_r) \rightarrow send(p_j, m_l) \Rightarrow delivery(p_j, m_r) \rightarrow send(p_j, m_l)$ , then  $bit_k \notin \Phi(p_j)$  and therefore,  $bit_k \notin h(m_l)$ , which is a contradiction.
- In the case where  $p_r = p_j$ , we have that  $delivery(p_j, m_k) \rightarrow delivery(p_j, m_r) \rightarrow send(p_j, m_l)$ , because the sending of  $m_r$  ( $send(p_j, m_r)$ ) takes place,  $bit_k$  is deleted from  $\Phi(p_j)$  by Line 1, 5 ( $\Phi(p) \leftarrow \emptyset$ ). Therefore, we have that  $bit_k \notin h(m_l)$ , which is a contradiction.

Finally, the following proposition shows that through the bits attached to the sent messages and the causal information stored at the base stations, we ensure the causal order in the mobile distributed system.

**Table 5.** Causal algorithms for mobile distributed systems without unnecessary inhibition in the message delivery.

Protocol	Communication Overhead (bytes)		Types of communication	Storage overhead (bytes)	
	wireless network	wired network		Base Stations	Mobile Hosts
AV-1	0	$c * n^2$ (always)	Unicast	$c * (n * n^2 + n)$ (always)	0
PRS	0	$c * n^2$ (worst case)	Multicast	$c * (n^2 + n^3 + n * s)$ (worst case)	0
Dependency Sequences	0	$c * (s * \epsilon)$ (not bounded)	Unicast	$c * (n * (s * \epsilon))$ (not bounded)	0
Mobi_Causal	0	$c * (\sum_{i=1}^s l_i)$ (always)	Unicast	$c * (n *  LastRcv )$ (not bounded)	0
HierarchicalClocks	0	$c * s$ (always)	Unicast	$c * (s + \phi)$ (not bounded)	0
MOKA	$n/b$ (worst case)	$c * 2n$ (worst case)	Group communication (broadcast case)	$c * 2n$ (worst case)	$n/b$ (worst case)

Where  $n$  = number of MHs,  $s$  = number of BSs,  $c$  is the number of bytes used to represent a integer value,  $b$  is the number of bits used to represent a byte,  $k$  is a predetermined integer parameter,  $\epsilon$  is the length of the longest dependency sequence for a MH,  $l_i$  represents the number of messages sent by base station BS <sub>$i$</sub> , and for which the delivery is not yet confirmed and  $LastRcv$  is a control information structure that stores identifiers of messages received by a MH.  
doi:10.1371/journal.pone.0059904.t005

**Proposition 1.**

$$bit_k \in h(m_l) \Rightarrow (i, a) \in H(m_l)$$

Proof. By Line 3, 20, we have that  $(i, a, k') \in CI(BS)$  only after the delivery of message  $m_k = (i, a, mes\_received(p_j), event, h(m_k))$  at the local base station  $BS$ . In this case,  $k'$  (by line 3, 14) 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 2, 1–5) that  $k' = mes\_received(p_j)$  and by Line 2, 8, we have that  $bit_k \in \Phi(p)$ . We know by lemma 2 that if  $bit_k \in h(m_l)$  then  $m_k \downarrow m_l$ . On the reception of message  $m_l$  sent by  $p_j$  with  $m_l = (p_j, b, mes\_received(p_j), event, h(m_l))$  at the  $BS$ , by Lines 3, 7–13, we have  $(i, a) \in H(m_l)$  because there is in  $CI(BS)$  an element  $(i, a, k')$  where  $k' = mes\_received(p_j)$ .

**Discussion**

We compare our protocol with the related work according to four aspects: message overhead sent over wireless/wired communication channels, storage overhead, unnecessary inhibition in the message delivery, and handoff complexity (see Table 5 and 6).

**Message and storage overhead and unnecessary inhibition in the message delivery**

In order to reduce the overhead sent over wireless communication channels, the protocols AV-2 [13], AV-3 [13], YHH [14], LH [5], and KHC [7], ensure causal ordering at and according to the Base Stations. However, these protocols give rise to two main problems. First, the causal order seen by the MHs greatly differ from the causal orders of messages in which they were originally sent. Secondly, they introduce unnecessary inhibition at the message delivery. This unnecessary inhibition is due to the serialization of messages at the BSs level, since a base station is unable to detect mutual concurrency between messages occurring at different MHs within a single cell.

In contrast, the protocols AV-1 [13], PRS [15], Dependency sequences [4], Hierarchical clocks [4], Mobi\_Causal [9], and our protocol, MOKA, maintain causal ordering explicitly among MHs. Hence, unnecessary inhibition never occurs. Nevertheless, the protocols AV-1, PRS, Dependency sequences and Mobi\_Causal highly increase the messages' overhead sent over the wired

communication channels and the storage overhead at base station, see Table 5.

The communication and storage overhead for these protocols is as follows. AV-1 attaches a matrix of size  $n * n$  to messages sent over the wired network. Therefore, the protocol AV-1 generates a constant communication overhead over the wired network of size  $c * n^2$  bytes where  $c$  is the number of bytes used to represent an integer value and  $n$  is the number of mobile hosts; and in order to achieve the causal ordering AV-1 needs a storage overhead of size  $c * (n * (n^2 + n))$  bytes. For the PRS, the communication overhead in the wired channel is dynamic having a worst case of size  $c * n^2$  bytes. We note that, in this protocol, the updating process of the control structures considers the acknowledgement by the mobile hosts for each message received to its local base station which increases the delay in the communication. Moreover, the storage overhead of PRS in the worst case at base station is of size  $c * (n^2 + n^3 + n * s)$  bytes where  $s$  is the number of base stations.

The work of Dependency Sequences attach per messages in the wired network an overhead of size  $c * (s * \epsilon)$  bytes, where  $\epsilon$  is the length of the longest dependency sequence for a MH, see Table 5. In order to bound the size of  $\epsilon$ , Prakash and Singhal [4] propose periodically using global checkpoints; however the global checkpoint is an expensive operation. In addition, the storage overhead of DS at base station is of size  $c * (n * (s * \epsilon))$  bytes, see Table 5. Next, for Mobi\_causal the overhead sent over the wired communication channels is equal to  $c * (\sum_{i=1}^s l_i)$  bytes, where  $l_i$  represents the number of messages sent by BS <sub>$i$</sub>  and for which the delivery is not yet confirmed. In the worst case, this number can be equal to  $n$ . Another drawback of Mobi\_causal is the unbounded growth of control information stored ( $LastRcv$ ) on each  $BS$  in order to achieve the causal ordering.

The work of Hierarchical clocks only sends overhead over the wired communication channels of size  $c * s$  bytes; nevertheless, the identification of causal predecessors of an event involves the evaluation of a recurrence relation which imposes high communication and computation overheads. This protocol uses a hierarchical clock,  $\phi$ , which is composed by a vector  $\phi^m$  and a bits vector  $\phi^i$  of a variable length, where the size of  $\phi^i$  is not bounded, see Table 5. The author as for the work of Dependency



**Table 6.** Comparisons of handoff complexity among all protocols.

Protocol	Number of sent messages between BSs (source and target)	Message size (bytes)	Handoff management
AV-1	1	$c*n^2$	Synchronous
Dependency Sequences	1	$c*(s*\epsilon)$ (not bounded)	
Hierarchical Clocks	1	$c*s$	
Mobi_Causal	1	$c*n$	
MOKA	1	$c*n$ (worst case)	Asynchronous

Where  $n$  = number of MHs,  $s$  = number of BSs,  $c$  is the number of bytes used to represent a integer value,  $k$  is a predetermined integer parameter,  $\epsilon$  is the length of the longest dependency sequence for a MH and  $l_i$  represents the number of messages sent by base station  $BS_i$  and for which the delivery is not yet confirmed.  
doi:10.1371/journal.pone.0059904.t006

Sequences proposes the use of global checkpoints in order to bound the size of the bits vectors.

In our proposal, the MOKA protocol, the size of the control information over the wired network depends on the number of concurrent messages that immediately precede a message  $m$ . Since  $H(m)$  has only the most recent messages that precede a message  $m$ , the overhead per message in the MOKA protocol to ensure causal ordering is given by the cardinality of  $H(m)$ , which can fluctuate between 0 and  $n$ . Therefore, the communication overhead in the wired channel is dynamic having a worst case of size  $c * 2n$  bytes. On the other hand, in our protocol, the control information attached to messages sent over the wireless network and stored at a mobile host is given by the cardinality of  $h(m)$ , where  $h(m)$  is a structure of bits. Again, the size of  $h(m)$  depends on the number of concurrent messages that immediately precede to a message. Therefore, the communication overhead in the wireless channel is in the worst case of size  $n/b$  bytes, where  $b$  is the number of bits used to represent a byte.

On the other hand, in our protocol the storage overhead at MH is of size  $n/b$  bytes and at base station is in the worst case of size  $c * 2n$  bytes. We notice that in our protocol, as for the minimal causal algorithm in [10], 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  $k$  concurrent messages occur decreases inversely proportional to the size of the communication group. This behavior has been shown in [10].

### Handoff complexity

Handoff complexity indicates the amount of causal information exchanged between base stations during the handoff module execution [7], see Table 6. Here we only analyze the handoff module of the works that do not have unnecessary inhibition at the message delivery. The handoff complexity is determined by two aspects: 1) the number of sent messages between BS's and 2) the size of messages. Thus, AV-1 needs to send a message of size  $c * n^2$  bytes when a MH moves to its new cell, see Table 6.

### References

- Tarkoma S (2003) Distributed event dissemination for ubiquitous agents. In Proceedings of the 10th ISPE International Conference on Concurrent Engineering. Madeira, Portugal. pp. 105–110.
- Ghim Y, Ah-Hwee T, Hwee-Hwa P (2007) Discovering and Exploiting Causal Dependencies for Robust Mobile Context-Aware Recommenders. IEEE Transactions on Knowledge and Data Engineering 19: 977–992.
- Pomares HSE, Estudillo J, Morales LA, Rodriguez GG (2009) An Intermedia synchronization mechanism for multimedia distributed systems, International Journal of Internet Protocol Technology, 4: 207–218.
- Prakash R, Singhal M (1997) Dependency sequences and hierarchical clocks: efficient alternatives to vector clocks for mobile computing systems. Journal of Wireless Networks 3: 349–360.
- Li C, Huang L (1999) A mobile support station based causal multicast algorithm in mobile computing environment. Proc. 11th Int'l Conf. info. Networking 23: 100–110.
- Skawratanonond C, Mittal N, Garg K (1999) A Lightweight Algorithm for Causal Message Ordering in Mobile Computing Systems. In: Proc. of 12th ISCA Intl. Conf. on Parallel and Distributed Computing Systems (PDCS). Florida, USA. pp. 245–250.
- Kuang C, Li Y, Chao T, Lu H (2000) A Causal Multicast Protocol for Mobile Distributed systems. IEICE TRANS. INF. & SYST. E83-D: 2065–2073.
- Chandra P, Kshemkalyani AD (2004), Causal Multicast in Mobile Networks. In: Proceedings of the IEEE Computer Society's 12th Annual International

The handoff module proposed by the Dependency sequences, the Hierarchical clocks and the Mobi\_causal needs a message of size  $c * (s * \epsilon)$ ,  $c * s$  and  $c * n$  bytes, respectively, where  $\epsilon$  is the length of longest dependency sequence for a MH, see Table 6. The main drawback of all these protocols is that during the handoff module execution, the other MHs cannot send messages until the handoff process concludes which inhibits the system execution and degrades the application performance.

Our protocol MOKA is the only asynchronous, and it only needs to send one message in the worst case of size  $c * n$  bytes. This is because the control message sent is ensured to be causally delivery along the causal messages of the MDS.

### Conclusions

The MOKA protocol has been presented. This protocol ensures the causal ordering according to the causal view of the mobile host, eliminating the inhibition effect in the message delivery. The causal protocol presented satisfies the MDS requirements since at the mobile hosts a low computational cost is needed because only binary operations and simple sums are used. Moreover, low memory buffer is used since only a structure of bits is stored. In addition the MOKA protocol is efficient in the overhead attached per message at the wired and the wireless communication channels. The overhead sent per message is characterized by being dynamically adapted according to the behavior of the concurrent messages. Finally, the handoff management process presented is characterized by an asynchronous execution, which allows for the transfer of a MH from one cell to another without the need to suspend the sending or delivery of new causal messages.

### Author Contributions

Conceived and designed the experiments: ELD SEPH. Performed the experiments: ELD SEPH. Analyzed the data: ELD SEPH MAM. Wrote the paper: ELD SEPH GRG MAM.

- Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications System. Vollandam, The Netherlands. pp. 213–220.
9. Benzaid C, Badache N (2005) *Mobi\_Causal: A protocol for Causal Message Ordering in Mobile Computing Systems*. *Mobile computing and communications Review* 9: 19–28.
  10. Pomares HSE, Fanchon J, Drira K (2004) *The Immediate Dependency Relation: An Optimal Way to Ensure Causal Group Communication*. *Annual Review of Scalable Computing* 6: 61–79.
  11. Lamport L (1978) *Time Clocks and the Ordering of Messages in Distributed Systems*. *Communications ACM* 21: 558–565.
  12. Mattern F (1988) *Virtual Time and Global States of Distributed Systems*. *Proc. Workshop on Parallel and Distributed Algorithms*. Chateau de Bonas, France. pp. 215–226.
  13. Alagar S, Venkatesan S (1997) *Causal Ordering in Distributed Mobile Systems*. *IEEE Transactions on Computers* 46: 353–361.
  14. Yen L, Huang L, Hwang S (1998) *A protocol for causally ordered message delivery in mobile computing systems*. *Journal of ACM/Baltzer Mobile Networks and Applications* 2: 365–372.
  15. Prakash R, Raynal M, Singhal M (1996) *An Efficient Causal ordering Algorithm for Mobile Computing Environments*, *Proceedings of the 16th International Conference on Distributed Computing Systems*. Washington, DC, USA. pp. 744–751.