Approximation Algorithm for Estimating Distances in Distributed Virtual Environments

Olivier Beaumont, Tobias Castanet, Nicolas Hanusse, Corentin Travers

Approximation Algorithm for Estimating Distances in Distributed Virtual Environments

Olivier Beaumont[1,2], Tobias Castanet[1], Nicolas Hanusse[1,3], and Corentin Travers[1,4]

[1] LaBRI, 351, cours de la Libération F-33405 Talence cedex France
[2] Inria Bordeaux - Sud-Ouest, 200, avenue de la Vieille Tour 33405 Talence cedex, France
[3] CNRS
[4] Bordeaux INP, Avenue des facultés, 33405 Talence

Abstract. This article deals with the issue of guaranteeing properties in Distributed Virtual Environments (DVEs) without a server. This issue is particularly relevant in the case of online games, that operate in a fully distributed framework and for which network resources such as bandwidth are the critical resources. Players typically need to know the distance between their character and other characters, at least approximately. They all share the same position estimation algorithm but, in general, do not know the current positions of others. We provide a synchronized distributed algorithm $\mathcal{A}_{lc}$ to guarantee, at any time, that the estimated distance d_{est} between any pair of characters A and B is always a $1 + \varepsilon$ approximation of the current distance d_{act}, regardless of movement pattern, and then prove that if characters move randomly on a n-dimensional grid, or follow a random continuous movement on up to three dimensions, the number of messages of $\mathcal{A}_{lc}$ is optimal up to a constant factor. In a more practical setting, we also show that the number of messages of $\mathcal{A}_{lc}$ for actual game traces is much less than the standard algorithm sending actual positions at a given frequency.

Keywords: Distributed Virtual Environments · Online games · Random walks · Distributed approximation algorithms · Peer-to-peer algorithms

1 Introduction

1.1 Context

The term *Distributed Virtual Environment* (DVE) refers to systems where geographically distant users, or players, participate in a highly interactive virtual world. The main examples of DVEs are online games, where players control characters that interact with each other, and may modify the shared environment. Usually, interactions between characters and/or objects of the environment are enabled when they are sufficiently close in the virtual world. For simplicity, in the rest of the paper, we will use *player* to denote both the player and the character.

The main difference between a DVE and a classical distributed system like a database, is that the states of objects in the virtual environment evolve even without changes issued by the users [15] since non-player characters go about their programmed activities, and objects must respect the physics of the game. Moreover, the amount of inputs per time unit is generally high, as players interact a lot with the environment.

DVE participants need to know the state of the virtual world, in order to display it correctly and to be able to interact with it. The two central aspects that need to be optimized in a DVE are consistency and responsiveness. Inconsistencies arise when two users see different versions of the virtual world. This is particularly problematic in recent games, where players often communicate with each others using voice communication programs, making inconsistencies more noticeable. On the other hand, responsiveness, the time interval between when a user executes an action (for example, pushing the button to shift gears) and when the effects of this action is perceived by the player (the car actually shifting gears), is unsatisfactory when this time delay is noticeable.

One difficulty is related to the number of exchanged messages. In general, increasing the number of communications between players contributes both to responsiveness (changes are transmitted earlier) and consistency (more messages allow a more accurate knowledge of the game's state). On the other hand, it has been shown in [14] that too many messages degrade network performance, leading to inconsistencies.

In practice, many games rely on a simple strategy, where players send updates at a regular rate to other players. The main flaw of this technique is a poor scalability in terms of bandwidth, as the number of messages increases quadratically with the number of players. Scalability is a concern for DVEs: some games are intended to be played by a large number of participants at the same time (e.g. MMORPGs such as World Of Warcraft). In addition, many online games are based on a client-server architecture. This has many disadvantages, as maintaining a server is often expensive, and exposes a single point of failure [17]. This leads to the incentive to study peer-to-peer solutions, where players share the role of the server among themselves, but in this context, bandwidth becomes crucial, as the network capacities of peers are usually lower than those of powerful servers. This article focuses on reducing bandwidth usage by limiting the number of exchanged messages. Several versatile techniques have been proposed to achieve this goal.

Data compression regroups techniques that can reduce bandwidth usage, but that are dependent on the application. For example Delta encoding [17], is an implementation trick where only differences between states are sent.

Dead-reckoning is a widely used tool, standardized in the Appendix E of [3]. Each player predicts the positions of the other players, extrapolating their movements after each update, typically based on their speed and acceleration.

Error induced by dead-reckoning can be measured by different means [4, 18], but Dead-reckoning aims at bounding *the additive error* on the players positions. The players know their own actual positions at any time, and for the other players, they only know estimated positions. Since all the players share the same estimation algorithm, each player is able to detect if the error on his/her own position as seen by another player is above a given threshold. When this happens, *the player sends a message to this player to correct the outdated estimated position*. Dead-reckoning is flexible, with regard to trade-offs between consistency, responsiveness and bandwidth usage [10], because increasing the Dead-reckoning threshold generates less communications, but lowers the accuracy of information, and vice versa. Research on dead-reckoning improved bandwidth usage mainly in two ways : get the best prediction possible [11], or improve the update policies (a survey on different update policies is given in [16]). This last aspect often relies on Interest Management.

Interest Management consists in filtering updates in order to send them only to players who might be interested. Different types of interest management are identified in [6,13]. Some application-specific approaches may also use the fact that human attention is limited, as in [5], where a set of five interesting players is defined at any given time, in order to send frequent updates to those players, but much less to other players.

Combinations of all these techniques can be used. In [7], an area of interest, similar to aura interest management, is used to modify the Dead-reckoning threshold.

In the context of interest management, estimating distances between players is very useful, as a player is rarely interested in knowing the exact state of far away objects. In addition, in some application-specific cases, distance may be important, for example when implementing a spell that heals all allies within a certain range. To the best of our knowledge, no distributed algorithm has been proposed to solve the problem of estimating the distance between users of a DVE. The objective of this paper is *to provide a solution allowing players to estimate the distances between them, with a condition on the relative error, while guaranteeing that the use of bandwidth is as small as possible*. In particular, it has to be bounded against an ideal algorithm that would send a minimum number of messages, based on a perfect knowledge of the game's state.

We identify two main articles related to this objective.

In [15], two techniques are proposed. First, *local-lag* reduces short-term inconsistencies, at the cost of less responsiveness: a delay between the time an operation is issued and the time when the operation becomes effective is added. Secondly, `timewarp` is proposed, an algorithm to ensure consistency. In this algorithm, each player remembers all previous operations and the time at which they were issued. If an operation is received by a player too late, the player rewinds the state of the world, immediately recomputing the current state, using all needed operations. These operations are user initiated, thus, the number of messages is proportional to the number of players, and to the length of time.

In [12], Dead-Reckoning is used to compensate for latencies and message losses on the network. TATSI, the average spatial error on players' positions over a time interval, is estimated with no latency or loss of message. Then, under the assumption of a constant acceleration, latencies and message losses are added to the model, and it is shown that the same TATSI can be obtained by lowering the dead-reckoning threshold (thus making DVE nodes send more messages than without latency and message losses).

To summarize, solutions from the literature are very consuming in term of messages and/or target an *additive bound* on the error. By contrast, this paper focuses on bounding the *relative error* on distances and keeping the number of message exchanges low.

1.2 Contribution

In terms of optimality in number of messages, Dead-reckoning is optimal for position estimation. Indeed, when using Dead-reckoning, players know where other players see them. Thus, a player sends updates if and only if the tolerated error between his/her actual position and his/her estimated position is exceeded, making it an optimal bandwidth strategy. On the other hand, since no two players know the actual distance between them, none of them can determine the exact error over the estimated distance, making distance estimation a much harder problem.

We consider deterministic algorithms that allow each player to estimate, at any time, the distances between him/her and the other players, while having a guarantee on the error. Initially, each player knows the exact position of every other player. The metric we use is the relative error given in Equation 1, where, at each instant t, $d_{act}(t)$ denotes the actual distance between two players, and $d_{est}(t)$ denotes their estimated distance,

$$\text{relative error } = \frac{|d_{act}(t) - d_{est}(t)|}{d_{est}(t)}. \tag{1}$$

We make sure this error measurement never exceeds ε, the maximum tolerated relative error for any pair of players, while minimizing the number of exchanged messages.

That is, Equation 2 must always hold, for every pair of players,

$$(1-\varepsilon)d_{est}(t) < d_{act}(t) < (1+\varepsilon)d_{est}(t). \tag{2}$$

We propose an algorithm, called *local change* and denoted by $\mathcal{A}_{lc}$. It relies on the same underlying principle as Dead-reckoning, where position estimations are deterministic and each player computes his/her own position as seen by other players, using the same deterministic algorithm. In $\mathcal{A}_{lc}$, player Bob sends his actual position p_B to another player Alice as soon as the estimate $\widetilde{p_B}$ of the position of Bob as seen by Alice deviates too much from his actual position, more precisely *as soon as Equation 3 is violated*, where d denotes the distance between two points. In addition, Alice will *immediately respond to Bob by also sending her actual position*.

$$d\left(p_B(t),\ \widetilde{p_B}(t)\right) < d_{est}(t) \times \frac{\varepsilon}{2}. \tag{3}$$

To quantify the performance of our algorithm, we compare the number of messages against an oracle with a full knowledge of the current state of the game, called *ideal algorithm* and denoted by $\mathcal{A}_{id}$. In $\mathcal{A}_{id}$, an exchange of messages happens only when, and as soon Equation 2 is violated.

Our results are threefold. First, without any assumption on how players move, we prove that with $\mathcal{A}_{lc}$, when there is no latency, the maximal error is never overcome: Equation 2 is always satisfied (Theorem 1, Section 2).

Secondly, in the case where movement is limited to the random part based on players' actions, which cannot be anticipated by the deterministic prediction algorithm, we prove that, given ε, $\mathcal{A}_{lc}$ is optimal in terms of number of message exchanges up to a constant factor. In sections 3, 4 and 5, we use two different movement patterns, both of which consisting, at each instant $t \in \mathbb{N}$, to chose a new position at a distance at most 1 from the last position.

Finally, this theoretical analysis is complemented by experiments in Section 6. We first perform experiments on synthetic traces. Then, we use actual traces from Heroes of Newerth [1], to compare $\mathcal{A}_{lc}$ with a *fixed frequency algorithm*, denoted by $\mathcal{A}_{ff}$. $\mathcal{A}_{ff}$ is commonly used in practice in online games, and sends updates periodically, by waiting w time units between updates. We show that overall, $\mathcal{A}_{lc}$ behaves better while never exceeding the maximal tolerated error.

In summary, the performance (without latency) of $\mathcal{A}_{id}, \mathcal{A}_{lc}, \mathcal{A}_{ff}$ and `timewarp` [15] are shown in the following table:

	number of messages	maximal error	number of violations
$\mathcal{A}_{id}$	$m_{id} \leq Tn(n-1)$	$\leq \varepsilon$	0
$\mathcal{A}_{lc}$	$O(m_{id})$	$\leq \varepsilon$	0
$\mathcal{A}_{ff}$	$\frac{T}{w}n(n-1)$	0 if $w = 1$ unbounded otherwise	$\Theta(Tn^2)$
`timewarp`	$O(Tn^2)$	0	0

T denotes the duration of the experiment, and n the number of participants in the DVE. We consider as a reference m_{id}, the (perfect knowledge based) number of messages sent by $\mathcal{A}_{id}$. In the worst case, $\mathcal{A}_{id}$ would make players send one message each instant (when movement is large compared to the distance), thus $m_{id} \leq Tn(n-1)$. Note that `timewarp` functions slightly differently than the others: it is intended to ensure strict consistency. The *number of violations* counts, over T time units, the number of distance pairs for which the error is above ε.

2 Model and Algorithms

Model: Let us first assume that $\varepsilon \in]0;1[$. Indeed, $\varepsilon = 0$ means that no error is tolerated, while $\varepsilon = 1$ would accept any estimate on the distance, provided it is larger than half the actual distance, which is not very informative. Since $\mathcal{A}_{lc}$ must enforce that Equation 3 holds true for any pair of players, we focus on two players Alice and Bob. We assume that the communication channel connecting them is without message loss nor latency, that local computations do not take time and that all players share a synchronized clock. At any instant $t \in \mathbb{N}$, let us denote the positions of both players as $p_A(t)$ and $p_B(t)$. A position is a vector whose dimension depends on the virtual world (for example, for a 3D world, a position is described by a vector in $\mathbb{N}^3$, or $\mathbb{R}^3$ in the case of continuous moves). Each player knows his/her own actual position, but may not know exactly where the other player is. These positions can change unpredictably, through the actions of users.

In Section 3 and Section 4, we conduct analyses on Random Walk (see below), in 1D and in up to 3D respectively. In Section 5, we use the Continuous Movement. As these movements are random, the best possible estimation of the position of other players is to assume they remain still, so that a player will estimate that the other players are at their last known position.

Random Walk is a discrete movement taking place on a n-dimensional grid. Thus, positions can be represented as values from $\mathbb{Z}^n$. If at instant $t \in \mathbb{N}$, a player following such movement is at position $p = (p_1, p_2, \ldots, p_n)$ he/she has $2n$ neighbors: $(p_1 - 1, p_2, \ldots, p_n)$, $(p_1 + 1, p_2, \ldots, p_n)$, $(p_1, p_2 - 1, \ldots, p_n)$, etc. The movement consists, at each instant, to choose one of the neighbors, each one having probability $\frac{1}{2n}$ to be chosen.

Continuous Movement consists at each instant, to select a value smaller than one, and to add a vector of norm equal to this value, and with a direction randomly chosen. **In 1D**, a moving player adds at each instant, a random number following a uniform

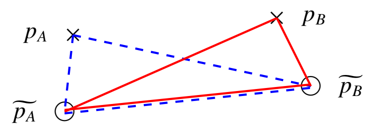


Fig. 1: Knowledge of Alice (dashed blue lines) and Bob (continuous red lines)

distribution on $[-1, 1]$ to their position. **In 2D**, at each instant t, a moving player X chooses ρ_t and θ_t following uniform distributions respectively on $[0, 1]$ and $[0, 2\pi]$, so that $p_X(t+1) = p_X(t) + (\rho_t, \theta_t)$, where (ρ_t, θ_t) is the vector with polar coordinates ρ_t and θ_t. **In 3D**, at each instant t, a moving player chooses ρ_t, θ_t, and φ_t following uniform distributions respectively on $[0, 1]$, $[0, 2\pi]$ and $[0, \pi]$.

Algorithm: As explained in Section 1.2, players will estimate their distance to each other. To do this, each player will compute a deterministic estimation of the other player's position, in order to get $d_{est}(t)$, i.e. Bob computes $\widetilde{p_A}(t)$, the estimate of the position of Alice, and Alice computes $\widetilde{p_B}(t)$. As they use the same deterministic algorithm, these computations can be replicated, and $\widetilde{p_A}(t)$ and $\widetilde{p_B}(t)$ become a shared knowledge, as seen on Figure 1 (even without communication). Thus, we will use the distance between those two (estimated but shared) positions as distance estimate, $d_{est}(t)$. In practice, $\widetilde{p_A}(t)$ is generally based on an extrapolation of Alice's position, speed and acceleration, from the time of the last message exchanged between Alice and Bob.

As explained in Section 1.2, $\mathcal{A}_{lc}$ sends updates of the actual position as soon as Equation 3 is not satisfied, as depicted in Algorithm 1. The other algorithm $\mathcal{A}_{id}$, used as a basis for comparison, sends updates as soon as the target inequality (Equation 2) becomes false, as depicted in Algorithm 2.

In Theorem 1, we prove that $\mathcal{A}_{lc}$ satisfies Equation 2, thus its correctness is established.

Theorem 1. *Using $\mathcal{A}_{lc}$, Equation 2 holds true at any instant (regardless of movement).*

Proof. The following inequalities hold true:

$$\begin{cases} d_{act}(t) - d_{est}(t) \leq d(p_A(t), \widetilde{p_A}(t)) + d(p_B(t), \widetilde{p_B}(t)) & \text{(triangle inequality)} \\ d_{est}(t) - d_{act}(t) \leq d(p_A(t), \widetilde{p_A}(t)) + d(p_B(t), \widetilde{p_B}(t)) & \text{(triangle inequality)} \\ d(p_B(t), \widetilde{p_B}(t)) < \frac{\varepsilon}{2} d_{est}(t) & \text{(by construction)} \\ d(p_A(t), \widetilde{p_A}(t)) < \frac{\varepsilon}{2} d_{est}(t) & \text{(by construction)} \end{cases}$$

so that $|d_{act}(t) - d_{est}(t)| < \varepsilon d_{est}(t)$, which is equivalent to Equation 2. □

3 Competitive Analysis in the 1D Random Walk Case

In this section, we focus on the 1D case, where players move along the integer line. The performance of $\mathcal{A}_{lc}$ is measured by M, the number of message exchanges (a message

Algorithm 1 Local change ($\mathcal{A}_{lc}$), from the point of view of Alice

$\boldsymbol{p_A} \leftarrow$ Alice's initial position ▹ Actual position of Alice. This is a read-only input to the algorithm
$\widetilde{p_A} \leftarrow$ Alice's initial position ▹ Position of Alice, as estimated by Bob, the other player
$\widetilde{p_B} \leftarrow$ Bob's initial position ▹ Estimated position of Bob
$d_{est} \leftarrow d(\widetilde{p_A}, \widetilde{p_B})$ ▹ Estimated distance. Will always be equal to $d(\widetilde{p_A}, \widetilde{p_B})$
procedure CHECK_FOR_UPDATE ▹ to be called at each $t \in \mathbb{N}$, after movement
  if $d(\boldsymbol{p_A}, \widetilde{p_A}) \geq \frac{\varepsilon}{2} d_{est}$ **then**
    $\widetilde{p_A} \leftarrow \boldsymbol{p_A}$
    $d_{est} \leftarrow d(\widetilde{p_A}, \widetilde{p_B})$
    send message ($\boldsymbol{p_A}$, begin_update) to Bob
  end if
procedure RECEIVE_MESSAGE(position, type) from Bob ▹ to be called when receiving a message
  $\widetilde{p_B} \leftarrow$ position
  $d_{est} \leftarrow d(\widetilde{p_A}, \widetilde{p_B})$
  if type = begin_update **then** ▹ type distinction is to avoid infinite messages
    send message ($\boldsymbol{p_A}$, update_reply) to Bob
  end if

and its response counting as one) between two players using $\mathcal{A}_{lc}$, before the first message sent with $\mathcal{A}_{id}$. In this setting, our result that $\mathcal{A}_{lc}$ is optimal is formally stated in Theorem 2 and Theorem 3, by an upper bound on the expectation of M. Note that this upper bound does not hold for a worst-case analysis: M can be infinitely large if players come and go, far enough for $\mathcal{A}_{lc}$ to send messages regularly, but not far enough for $\mathcal{A}_{id}$ to send messages.

Let us denote by d_{est} and $\widetilde{p}$ the estimates for $\mathcal{A}_{lc}$. We will consider instants t_i (with $i \geq 1$), defined as the instants at which the i-th round trip of the messages is sent with $\mathcal{A}_{lc}$. Both t_i and M are discrete random variables.

Since the movements are 1D (and because $d(p_1, p_2) = |p_1 - p_2|$), the update conditions of $\mathcal{A}_{lc}$ and $\mathcal{A}_{id}$ can be represented by intervals. More precisely, $\mathcal{A}_{lc}$ generates a message exchange as soon as p_X leaves I_{lcX}, where I_{lcX} is defined as follows, and X is either Alice or Bob.

Definition 1. $\forall t \in [\![t_i; t_{i+1}[\![$, *then* $I_{lcX}(t) = \left]\widetilde{p_X}(t_i) - d_{est}(t_i)\frac{\varepsilon}{2} \ ; \ \widetilde{p_X}(t_i) + d_{est}(t_i)\frac{\varepsilon}{2}\right[$

Let $d_0 = d_{act}(0)$. Algorithm $\mathcal{A}_{id}$ generates a message as soon as d_{act} leaves I_{id}, where I_{id} is defined by $I_{id} =]d_0\,(1-\varepsilon) \ ; \ d_0\,(1+\varepsilon)[$. Let $t_{opt} = \min\{t : d_{act}(t) \notin I_{id}\}$ denote the time of the first message sent by $\mathcal{A}_{id}$, then

$$M = \max\{i,\ t_i \leq t_{opt}\}.$$

Let us now define the auxiliary random variable $M' : \min\{i,\ d_{est}(t_i) \notin I_{id}\}$. M' represents the index of the first message of $\mathcal{A}_{lc}$ sent after Bob left I_{id}. At this instant, by construction, $\mathcal{A}_{id}$ already sent a message. This is formally stated in the following proposition, which states that an upper bound for M' also holds for M.

Proposition 1. $M' \geq M$

Algorithm 2 The ideal algorithm, $\mathcal{A}_{id}$

$\boldsymbol{p_A} \leftarrow$ Alice's initial position ▷ Actual position of Alice. This is a read-only input to the algorithm
$\boldsymbol{p_B} \leftarrow$ Bob's initial position ▷ Actual position of Bob. This is a read-only input to the algorithm
$\widetilde{p_A} \leftarrow$ Alice's initial position (for both players) ▷ Estimated position of Alice
$\widetilde{p_B} \leftarrow$ Bob's initial position (for both players) ▷ Estimated position of Bob
$d_{est} \leftarrow d(\widetilde{p_A}, \widetilde{p_B})$ (for both players) ▷ Estimated distance
procedure CHECK_FOR_UPDATE ▷ to be called at each tick, after movement
    if $|d_{act}, d_{est}| \geq d_{est}\varepsilon$ **then**
        Both players send their positions:
        $\widetilde{p_A} \leftarrow \boldsymbol{p_A}$
        $\widetilde{p_B} \leftarrow \boldsymbol{p_B}$
        $d_{est} \leftarrow d(\widetilde{p_A}, \widetilde{p_B})$
    end if
end procedure

Proof. By definition of $\mathcal{A}_{lc}$, for every i, $d_{est}(t_i) = d_{act}(t_i)$. Thus, $t_{M'} \in \{t,\ d_{act}(t) \notin I_{id}\}$, so that $t_{M'} \geq t_{opt}$. Since $t_{opt} \geq t_M$, $t_{M'} \geq t_M$ and $M' \geq M$. □

3.1 Case when only one of the players moves

Let us start with the case when only one of the two players follows a 1D Random Walk, as described in Section 2. Then, $p_A(t) = 0$ at any time step and Bob moves on $\mathbb{N}$, starting at distance $d_0 > 0$ from Alice so that $p_B(0) = d_0$ and

$$p_B(t+1) = \begin{cases} p_B(t) + 1 \text{ with probability } \frac{1}{2} \\ p_B(t) - 1 \text{ with probability } \frac{1}{2}. \end{cases}$$

We assume that d_{est} remains constant between two message exchanges in $\mathcal{A}_{lc}$, i.e. $\forall t \in [\![t_i, t_{i+1}[\![$, $d_{est}(t) = d_{est}(t_i)$. As a result, we have the following proposition.

Proposition 2. *With 1D movements, $\forall t \in [\![t_i, t_{i+1}[\![$, $\mathcal{A}_{lc}$ triggers the $i+1$-th round trip of messages as soon as $p_B(t)$ gets out of $I_{lc}(t) = \left]d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right)\ ;\ d_{est}(t_i)\left(1 + \frac{\varepsilon}{2}\right)\right[$.*

Proof. Since Alice always remains at the origin, all messages are generated by Bob and $\forall t \in [\![t_i; t_{i+1}[\![$, $\widetilde{p_B}(t) = d_{est}(t)$. Moreover, since $d_{est}(t) = d_{est}(t_i)$, then for Bob, Equation 3 is equivalent to $|p_B(t) - d_{est}(t_i)| < d_{est}(t_i) \times \frac{\varepsilon}{2}$, which in turn is equivalent to $p_B(t) \in I_{lc}(t)$. Similarly, in the case of $\mathcal{A}_{id}$, the first message is sent as soon as p_B gets out of I_{id} as $p_B(t) = d_{act}(t)$. □

First upper bound on M We provide a first upper bound on the expected value of M, that does not depend on the initial distance between the players.

Theorem 2. *Let $\Delta_l = \left\lceil \frac{\log(1-\varepsilon) - \log(1+\varepsilon)}{\log(1-\frac{\varepsilon}{2})} \right\rceil$ and $\varepsilon \in]0;1[$. With two players, one of them following a random walk, on $\mathbb{Z}$, $\mathbb{E}[M] \leq \Delta_l \times 2^{\Delta_l}$.*

To prove this, let us first look at the estimated distance. When a message is sent in $\mathcal{A}_{lc}$, $d_{est}(t_{i+1})$ can take only two values, as stated in Proposition 3.

Proposition 3. $d_{est}(t_{i+1}) = \begin{cases} \left\lfloor d_{est}(t_i)\left(1-\frac{\varepsilon}{2}\right)\right\rfloor = d_{est}(t_i) - \left\lceil \frac{\varepsilon}{2} d_{est}(t_i)\right\rceil & \textit{(with probability } \frac{1}{2}\textit{)} \\ \left\lceil d_{est}(t_i)\left(1+\frac{\varepsilon}{2}\right)\right\rceil = d_{est}(t_i) + \left\lceil \frac{\varepsilon}{2} d_{est}(t_i)\right\rceil & \textit{(with probability } \frac{1}{2}\textit{)} \end{cases}$

Proof. By definition of $\mathcal{A}_{lc}$, and since positions of Bob are integers, a message is sent when the position of Bob gets to the first integer position outside of I_{lc}. The rightmost equalities directly follow the properties of floor and ceiling function. Thus, the two possible positions at time t_{i+1} are at a same distance from $d_{est}(t_i)$ and have therefore the same probability. □

As a result, $d_{est}(t_{i+1})$ can only take two different values depending on $d_{est}(t_i)$, both having the same probability. We will call *jump*, and denote by m_i the transformation between $d_{est}(t_i)$ and $d_{est}(t_{i+1})$, where

$$m_i = \begin{cases} l \text{ if } d_{est}(t_{i+1}) = l(d_{est}(t_i)) \text{ where } l : x \mapsto \left\lfloor x\left(1-\frac{\varepsilon}{2}\right)\right\rfloor \\ r \text{ if } d_{est}(t_{i+1}) = r(d_{est}(t_i)) \text{ where } r : x \mapsto \left\lceil x\left(1+\frac{\varepsilon}{2}\right)\right\rceil \end{cases} \quad (4)$$

We can now prove Lemma 1 which states that, if there are enough successive l-jumps, then Bob will get out of I_{id}, whatever his initial position in the interval I_{id}.[5]

Lemma 1. *For all* $x \in I_{id}$*,* $l^{\Delta_l}(x) \leq d_0(1-\varepsilon)$*.*

Proof. $x \in I_{id} \Rightarrow x \leq d_0(1+\varepsilon) \Rightarrow l^{\Delta_l}(x) \leq l^{\Delta_l}(d_0(1+\varepsilon))$ since l is increasing, implying that $l^{\Delta_l}(x) \leq d_0(1+\varepsilon)\left(1-\frac{\varepsilon}{2}\right)^{\Delta_l}$ since $\forall x, l(x) \leq x\left(1-\frac{\varepsilon}{2}\right)$. Moreover, since, $\Delta_l \geq \frac{\log(1-\varepsilon)-\log(1+\varepsilon)}{\log\left(1-\frac{\varepsilon}{2}\right)}$ and $\log\left(1-\frac{\varepsilon}{2}\right) < 0$, then $(1+\varepsilon)\left(1-\frac{\varepsilon}{2}\right)^{\Delta_l} \leq (1-\varepsilon)$ and $x \in I_{id} \Rightarrow l^{\Delta_l}(x) \leq d_0(1-\varepsilon)$ □

Proof. Let us now prove Theorem 2. Let us split the sequence of movements of Bob into *phases* of length Δ_l and let us denote by j the index of the phase containing jumps from $m_{(j-1)\Delta_l}$ to $m_{j\Delta_l-1}$. Let us consider the following possible events (i) $\mathcal{S}_j$: there is at least one $i \in [\![(j-1)\Delta_l; j\Delta_l]\!]$ such that $d_{est}(t_i) \notin I_{id}$ and (ii) $\mathcal{S}'_j$: phase j is composed of l-jumps only. In turn, these events can be used to define useful random variables: (i) $X_j = 1$ if $\mathcal{S}_j$ is true, 0 otherwise (ii) $X'_j = 1$ if $\mathcal{S}'_j$ is true, 0 otherwise, (iii) $Y = j$ if $X_j = 1$ and $X_k = 0$ for every $k < j$ and (iv) $Y' = j$ if $X'_j = 1$ and $X'_k = 0$ for every $k < j$. Thus, Y denotes the index of the first phase during which Bob gets out of I_{id}.

If $\mathcal{S}'_j$ is true, then $d_{est}(t_{j\Delta_l}) = l^{\Delta_l}(d_{est}(t_{(j-1)\Delta_l}))$. Thus, by Lemma 1, $\mathcal{S}'_j \Rightarrow \mathcal{S}_j$, so that $X'_j = 1 \Rightarrow X_j = 1$.

$$\text{Therefore } Y' = j \Rightarrow X'_j = 1 \Rightarrow X_j = 1 \Rightarrow Y \leq j \text{ and finally } \mathbb{E}[Y] \leq \mathbb{E}[Y'] \quad (5)$$

Moreover, we know that Y' follows a geometric distribution with parameter $\mathbb{P}(\mathcal{S}'_j) = \frac{1}{2^{\Delta_l}}$ (because each jump has a $\frac{1}{2}$ probability of being l or r), so that $\mathbb{E}[Y'] \leq 2^{\Delta_l}$. Thus, by Equation 5, we have $\mathbb{E}[Y] \leq 2^{\Delta_l}$. Since Y denotes the index of the first phase during which Bob gets out of I_{id}, $M' \in [\![(Y-1)\Delta_l; Y\Delta_l]\!]$. In particular, $M' \leq Y\Delta_l$ and $\mathbb{E}[M'] \leq \Delta_l \times 2^{\Delta_l}$. Finally, Proposition 1 proves that $\mathbb{E}[M] \leq \Delta_l \times 2^{\Delta_l}$. □

[5] NB: we could have used r-jumps, but values are better with l-jumps.

Second upper bound on M

Theorem 3. *Let Δ_l be defined as previously. If $\varepsilon \in]0;1[$, with two players, one of them following a random walk, on $\mathbb{Z}$, then* $\mathbb{E}[M] \leq \left\lceil \frac{4}{\pi} {\Delta_l}^2 \right\rceil \times 8$

We provide a tighter analysis for M, which is formally stated in Theorem 3. To establish this result, we no longer consider phases consisting only of l-jumps, but also phases with a sufficient excess of l-jumps. This is because a sequence of an l-jump and a r-jump (in any order) tends to reduce the distance, as proved in Proposition 4.

Let $m_{i,j} = m_{j-1} \circ m_{j-2} \circ \cdots \circ m_i$, so that $d_{est}(t_j) = m_{i,j}(d_{est}(t_i))$. We will need the following result in order to prove Theorem 3:

Theorem 4. *Let $\sigma = card(\{k, m_k = l, k \in [\![i, j-1]\!]\}) - card(\{k, m_k = r, k \in [\![i, j-1]\!]\})$ denote the excess in l from m_i to m_{j-1}. If $\sigma \geq \Delta_l$, and $x \in I_{id}$, then $m_{i,j}(x) \notin I_{id}$.*

But first, we will prove a few properties:

Proposition 4. *$\forall p \in \mathbb{N}$, $l \circ r(p) \leq p$, and $r \circ l(p) \leq p$.*

Proof. $l(p) = p - \left\lceil \frac{p\varepsilon}{2} \right\rceil$, and $r(p) = p + \left\lceil \frac{p\varepsilon}{2} \right\rceil$, so that

$$l \circ r(p) = p + \left\lceil \frac{p\varepsilon}{2} \right\rceil - \left\lceil \frac{p\varepsilon}{2} + \left\lceil \frac{p\varepsilon}{2} \right\rceil \frac{\varepsilon}{2} \right\rceil \leq p \text{ since } \left\lceil \frac{p\varepsilon}{2} \right\rceil \leq \left\lceil \frac{p\varepsilon}{2} + \left\lceil \frac{p\varepsilon}{2} \right\rceil \frac{\varepsilon}{2} \right\rceil$$

$$\text{and } r \circ l(p) = p - \left\lceil \frac{p\varepsilon}{2} \right\rceil + \left\lceil \frac{p\varepsilon}{2} - \left\lceil \frac{p\varepsilon}{2} \right\rceil \frac{\varepsilon}{2} \right\rceil \leq p \text{ since } \left\lceil \frac{p\varepsilon}{2} - \left\lceil \frac{p\varepsilon}{2} \right\rceil \frac{\varepsilon}{2} \right\rceil \leq \left\lceil \frac{p\varepsilon}{2} \right\rceil.$$

□

Proposition 5. *$\forall (p, q) \in \mathbb{N}^2, \forall s \in \mathbb{N}$ and $\forall f = f_1 \circ f_2 \circ \cdots \circ f_s$, where $f_k = l$ or r for all $1 \leq k \leq s$, if $p \leq q$, then $f(p) \leq f(q)$.*

Proof. The proof is obtained by noting that ceiling and the floor functions, l and r and their compositions are increasing functions. □

Lemma 2. *Let $j > i$ and let us assume that $\sigma = card(\{k : m_k = l, k \in [\![i, j-1]\!]\}) - card(\{k : m_k = r, k \in [\![i, j-1]\!]\}) \geq 0$, then $\forall p \in \mathbb{N}$, $m_{i,j}(p) \leq l^{\sigma}(p)$.*

Proof. Let $f = f_1 \circ f_2 \circ \cdots \circ f_s$ where $f_k = l$ or r for all $1 \leq k \leq s$. Let $T : f \mapsto f'$ with $f' = f_1 \circ \cdots \circ f_k \circ f_{k+3} \circ \cdots \circ f_s$ so that $f_{k+1} \circ f_{k+2} = r \circ l$ or $l \circ r$, i.e. T simply consists of removing the first occurrence of $r \circ l$ or $l \circ r$. Then, $f_{k+1} \circ f_{k+2} \circ f_{k+3} \circ \cdots \circ f_s(x) \leq f_{k+3} \circ \cdots \circ f_s(x)$ thanks to Proposition 4, and $f_1 \circ \cdots \circ f_s(x) \leq f_1 \circ \cdots \circ f_k \circ f_{k+3} \circ \cdots \circ f_s(x)$ thanks to Proposition 5, so that

$$f(x) \leq T(f)(x). \tag{6}$$

Let $T^* : f \mapsto f^*$ with f^* being the result of the recursive application of T on f until only ls remain (remember that $\sigma \geq 0$). By Equation 6, $f(p) \leq T^*(f)(p)$. As $T^*(m_{i,j}) = l^{\sigma}$, finally $m_{i,j}(p) \leq l^{\sigma}(p)$. □

Proof. Let us now prove Theorem 4. Let $\sigma \geq \Delta_l$ and $p \in I_{id}$. Since $\Delta_l > 0$, $\sigma > 0$. Thus, thanks to Lemma 2, $f_{i,j}(x) \leq l^{\sigma}(x) \leq l^{\Delta_l}(x)$(because $l(x) \leq x$ and $\sigma \geq \Delta_l$) $\leq d_0(1-\varepsilon)$ thanks to Lemma 1. Thus, by definition of I_{id}, we have $f_{i,j}(p) \notin I_{id}$ □

The following lemma provides a lower bound on the probability of the event $\sigma \geq \Delta_l$, that will be later use to upper bound the expectation of M.

Lemma 3. *If* $j - i = 2\left\lceil \frac{4}{\pi}{\Delta_l}^2 \right\rceil$, *then* $\mathbb{P}(\sigma \geq \Delta_l) \geq \frac{1}{4}$.

Proof. Let $\Phi = j - i = 2\left\lceil \frac{4}{\pi}{\Delta_l}^2 \right\rceil$ the number of jumps between m_i and m_{j-1}, and let $\Lambda = \text{card}(\{k : m_k = l, k \in [\![i, j-1]\!]\})$, the number of l-jumps between i and $j-1$. Then, $\sigma \geq \Delta_l \Leftrightarrow 2\Lambda - \Phi \geq \Delta_l \Leftrightarrow \Lambda \geq \frac{\Delta_l + \Phi}{2}$ so that

$$\mathbb{P}(\sigma \geq \Delta_l) = \mathbb{P}\left(\Lambda \geq \frac{\Delta_l + \Phi}{2}\right) = \sum_{k=\left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil}^{\Phi} \binom{\Phi}{k} \times \frac{1}{2^\Phi} \text{ because } \mathbb{P}(\Lambda = k) = \binom{\Phi}{k} \times \frac{1}{2^\Phi}$$

$$= \frac{1}{2^\Phi}\left(\sum_{k=\frac{\Phi}{2}+1}^{\Phi} \binom{\Phi}{k} - \sum_{k=\frac{\Phi}{2}+1}^{\left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil - 1} \binom{\Phi}{k}\right) \text{ because } \left\lceil \frac{\Delta_l + \Phi}{2} \right\rceil > \frac{\Phi}{2} + 1$$

Moreover, as Φ is even, $\sum_{k=0}^{\Phi}\binom{\Phi}{k} = 2^\Phi = 2 \times \sum_{k=\frac{\Phi}{2}+1}^{\Phi}\binom{\Phi}{k} + \binom{\Phi}{\frac{\Phi}{2}}$ so that

$$\mathbb{P}(\sigma \geq \Delta_l) = \frac{1}{2^\Phi}\left(\frac{2^\Phi - \binom{\Phi}{\frac{\Phi}{2}}}{2} - \sum_{k=\frac{\Phi}{2}+1}^{\left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil - 1} \binom{\Phi}{k}\right)$$

$$= \frac{1}{2} - \frac{\binom{\Phi}{\frac{\Phi}{2}}}{2^{\Phi+1}} - \frac{1}{2^\Phi} \sum_{k=\frac{\Phi}{2}}^{\left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil - 1} \binom{\Phi}{k} + \frac{\binom{\Phi}{\frac{\Phi}{2}}}{2^\Phi} \geq \frac{1}{2} - \frac{1}{2^\Phi} \sum_{k=\frac{\Phi}{2}}^{\left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil - 1} \binom{\Phi}{k}$$

There are $\left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil - \frac{\Phi}{2}$ elements in the remaining sum. Note that $\left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil \leq \frac{\Delta_l+\Phi}{2} + 1 \Rightarrow \left\lceil \frac{\Delta_l+\Phi}{2} \right\rceil - \frac{\Phi}{2} \leq \frac{\Delta_l}{2} + 1$ and that each element of the sum is smaller than the first one since $\binom{\Phi}{\frac{\Phi}{2}} \geq \binom{\Phi}{\frac{\Phi}{2}+n}$. Therefore,

$$\mathbb{P}(\sigma \geq \Delta_l) \geq \frac{1}{2} - \frac{1}{2^\Phi} \times \left(\frac{\Delta_l}{2} + 1\right) \times \binom{\Phi}{\frac{\Phi}{2}}$$

$$\geq \frac{1}{2} - \frac{1}{2^\Phi} \times \left(\frac{\Delta_l}{2} + 1\right)\left(\frac{2^\Phi}{\sqrt{\frac{\Phi}{2} \times \pi}}\right)$$

$$\geq \frac{1}{2}\left(1 - \frac{\Delta_l\sqrt{2}}{\sqrt{\Phi\pi}}\right) + \left(\frac{\sqrt{2}}{\sqrt{\Phi\pi}}\right)$$

$$\geq \frac{1}{2}\left(1 - \sqrt{\frac{\pi}{8}} \times \sqrt{\frac{2}{\pi}}\right) + \left(\frac{\sqrt{2}}{\sqrt{\Phi\pi}}\right) \text{ because } \Phi \geq \frac{8}{\pi}{\Delta_l}^2 \Rightarrow \sqrt{\frac{\pi}{8}} \geq \frac{\Delta_l}{\sqrt{\Phi}}$$

$$\geq \frac{1}{4} + \left(\frac{\sqrt{2}}{\sqrt{\Phi\pi}}\right) \geq \frac{1}{4} \text{ because } \Phi > 0.$$

□

Proof. We can now prove the main result of this section, i.e. Theorem 3. As for the proof of Theorem 2, let us split the sequence of Bob movements in *phases* of length Φ and let us denote by j the index of the phase containing jumps $m_{(j-1)\Phi}$ through $m_{j\Phi-1}$. Let us consider following events (i) $\mathcal{S}_j$: there is at least one $i \in [\![(j-1)\Phi; j\Phi]\!]$ such that $d_{est}(t_i) \notin I_{id}$ and (ii) $\mathcal{S}''_j$: either $d_{est}(t_{(j-1)\Phi}) \notin I_{id}$, or $d_{est}(t_{j\Phi}) \notin I_{id}$. These events can in turn be used to define the following random variables (i) $X_j = 1$ if $\mathcal{S}_j$ is true, 0 otherwise (ii) $X''_j = 1$ if $\mathcal{S}''_j$ is true, 0 otherwise (iii) $Y = j$ if $X_j = 1$ and $X_k = 0$ for every $k < j$ and (iv) $Y'' = j$ if $X''_j = 1$ and $X''_k = 0$ for every $k < j$. Thus Y denotes the index of the first phase during which Bob gets out of I_{id}.

If $\mathcal{S}''_j$ is true, then $\mathcal{S}_j$ also holds true. Thus, in a similar way as for Theorem 2, $Y'' = j \Rightarrow X''_j = 1 \Rightarrow X_j = 1 \Rightarrow Y \leq j$ and thus $\mathbb{E}[Y] \leq \mathbb{E}[Y'']$. Moreover, by Theorem 4 and Lemma 3, if $\Phi = \left\lceil \frac{4}{\pi}\Delta_l{}^2 \right\rceil \times 2$, then $\mathbb{P}(\mathcal{S}''_j) \geq \frac{1}{4}$. Note that Y'' follows a geometric distribution with parameter $\mathbb{P}(\mathcal{S}''_j)$, so that $\mathbb{E}[Y] \leq \mathbb{E}[Y''] \leq 4$. Since Y denotes the index of the first phase during which Bob gets out of I_{id}, then $M' \in [\![(Y-1)\Phi; Y\Phi]\!]$. In particular, since $M' \leq Y\Phi$, $\mathbb{E}[M'] \leq \Phi \times 4$. By Proposition 1, we get $\mathbb{E}[M] \leq \left\lceil \frac{4}{\pi}\Delta_l{}^2 \right\rceil \times 8$. □

Conclusion In the 1D case, we prove that $\mathbb{E}[M]$ is smaller than both $\Delta_l \times 2^{\Delta_l}$ and $\left\lceil \frac{4}{\pi}\Delta_l{}^2 \right\rceil \times 8$, where $\Delta_l = \left\lceil \frac{\log(1-\varepsilon)-\log(1+\varepsilon)}{\log(1-\frac{\varepsilon}{2})} \right\rceil$. Actually, the choice of the best upper bound depends on values of ε. We can also observe that $\lim_{\varepsilon \to 1} \Delta_l = \infty$, meaning that there is no upper bound on M when ε is close to 1. This is not surprising, since a value of 1 for ε would make the left bound of I_{id} become 0 and $\mathcal{A}_{lc}$ could perform an infinite number of l-jumps before the first message of $\mathcal{A}_{id}$ if d_0 is large enough. Experiments depicted in Section 6 indeed show that M can become large when ε gets close to one.

3.2 Case when both players move

In this section, we consider that both players move (under the same stochastic movement model) on the integer line $\mathbb{Z}$. Again, we concentrate on a single pair of players Alice and Bob but the results apply to any pair of players and therefore can be extended to any number of players.

At each instant t, Alice moves $p_A(t) = p_A(t+1) = \begin{cases} p_A(t-1)+1 \text{ with probability } \frac{1}{2} \\ p_A(t-1)-1 \text{ with probability } \frac{1}{2} \end{cases}$

and Bob moves too $p_B(t) = p_B(t+1) = \begin{cases} p_B(t-1)+1 \text{ with probability } \frac{1}{2} \\ p_B(t-1)-1 \text{ with probability } \frac{1}{2} \end{cases}$

The equality between position and distance ($\widetilde{p_B}(t) = d_{est}(t)$) is no longer valid so that Proposition 2 does not hold and we rely on Definition 1. The definition of interval I_{id} remains unchanged, and messages are exchanged at t such that $d_{act}(t) \notin I_{id}$ (and not p_B). Theorem 5 is an extension of Theorem 2 in the case where both players move.

Theorem 5. *Let Δ_l be defined as previously. If $\varepsilon \in]0;1[$, then with two players following a random walk on $\mathbb{Z}$, $\mathbb{E}[M] \leq \Delta_l \times 4^{\Delta_l}$*

Proof. Assume, without loss of generality that Bob remains to the right of Alice, that is, $p_B > p_A$. After the $(i+1)$-th round trip of messages in $\mathcal{A}_{lc}$, i.e. at instant t_{i+1}, one of the four following events takes place (i) $\mathcal{B}_l$: at instant t_{i+1}, player Bob gets out of I_{lcB} by getting closer to Alice; (ii) $\mathcal{B}_r$: at instant t_{i+1}, player Bob gets out of I_{lcB} by getting farther from Alice; (iii) $\mathcal{A}_l$: at instant t_{i+1}, player Alice gets out of I_{lcA} by getting farther from Bob ; (iv) $\mathcal{A}_r$: at instant t_{i+1}, player Alice gets out of I_{lcA} by getting closer to Bob.

At least one of these events has to be true: $\mathbb{P}(\mathcal{B}_l \cup \mathcal{B}_r \cup \mathcal{A}_l \cup \mathcal{A}_r) = 1$. Additionally, all four events have the same probability, as both players start at the center of their interval at instant t_i. Thus, $\mathbb{P}(\mathcal{B}_l) = \mathbb{P}(\mathcal{B}_r) = \mathbb{P}(\mathcal{A}_l) = \mathbb{P}(\mathcal{A}_r) \geq \frac{1}{4}$.

Let us consider for instance the situation where $\mathcal{B}_l$ is true, i.e. $\widetilde{p_B}(t_{i+1}) = \widetilde{p_B}(t_i) - \left\lceil d_{est}(t_i) \times \frac{\varepsilon}{2} \right\rceil$. When Bob gets out of $I_{lcB}(t_i)$, as movement is symmetric, Alice has one half probability to be on one side of $\widetilde{p_A}(t_i)$, thus $\mathbb{P}(\widetilde{p_A}(t_{i+1}) \geq \widetilde{p_A}(t_i) | \mathcal{B}_l) \geq \frac{1}{2}$.

Moreover, $\widetilde{p_A}(t_{i+1}) \geq \widetilde{p_A}(t_i) \Rightarrow \widetilde{p_B}(t_{i+1}) - \widetilde{p_A}(t_{i+1}) \leq \widetilde{p_B}(t_i) - \widetilde{p_A}(t_i) - \left\lceil d_{est}(t_i)\frac{\varepsilon}{2} \right\rceil$ by definition of $\mathcal{B}_l$. Therefore, we have that $d_{est}(t_{i+1}) \leq d_{est}(t_i) - \left\lceil d_{est}(t_i)\frac{\varepsilon}{2} \right\rceil$ because Bob is on the right side of Alice. Finally, we get $d_{est}(t_{i+1}) \leq d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right)$. Thus, $\mathbb{P}\left(d_{est}(t_{i+1}) \leq d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right) | \mathcal{B}_l\right) \geq \frac{1}{2}$ and by a comparable reasoning on player Alice, we get $\mathbb{P}\left(d_{est}(t_{i+1}) \leq d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right) | \mathcal{A}_r\right) \geq \frac{1}{2}$. Thus, using the law of total probability, we get $\mathbb{P}\left(d_{est}(t_{i+1}) \leq d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right)\right) \geq \frac{1}{2} \times \mathbb{P}(\mathcal{B}_l) + \frac{1}{2} \times \mathbb{P}(\mathcal{A}_r) + 0 \times \mathbb{P}(\mathcal{B}_r) + 0 \times \mathbb{P}(\mathcal{A}_l) \geq \frac{1}{4}$ Repeating this operation Δ_l times, we get $\mathbb{P}\left(d_{est}(t_{i+\Delta_l}) \leq d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right)^{\Delta_l}\right) \geq \frac{1}{4^{\Delta_l}}$. and similarly to Lemma 1, we get $d_{est}(t_i) \in I_{id} \Rightarrow d_{est}(t_i) \leq d_0(1+\varepsilon) \Rightarrow d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right)^{\Delta_l} \leq d_0(1+\varepsilon)\left(1 - \frac{\varepsilon}{2}\right)^{\Delta_l}$ and since $\Delta_l \geq \frac{\log(1-\varepsilon) - \log(1+\varepsilon)}{\log\left(1 - \frac{\varepsilon}{2}\right)}$ then $d_{est}(t_i) \in I_{id} \Rightarrow d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right)^{\Delta_l} \leq d_0(1-\varepsilon) \Rightarrow d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right)^{\Delta_l} \notin I_{id}$.

$$\text{Hence, } \forall i, \quad d_{est}(t_i) \in I_{id} \Rightarrow \mathbb{P}\left(d_{est}(t_{i+\Delta_l}) \notin I_{id}\right) \geq \frac{1}{4^{\Delta_l}} \tag{7}$$

To prove Theorem 5, we rely on the same techniques as for Theorem 2 and Theorem 3, by splitting the sequence of jumps into phases of length Δ_l, and by denoting by j the index of the phase containing jumps $m_{(j-1)\Phi}$ through $m_{j\Phi-1}$. Let us consider the event $\mathcal{S}_j$: there is at least one $i \in [\![(j-1)\Delta_l; j\Delta_l]\!]$ such that $d_{est}(t_i) \notin I_{id}$ and the random variables (i) $X_j = 1$ if $\mathcal{S}_j$ is true, 0 otherwise, (ii) $Y = j$ if $X_j = 1$ and $X_k = 0$ for all $k < j$. By Equation 7, if $d_{est}(t_{(j-1)\Delta_l}) \in I_{id}$, then $\mathbb{P}\left(d_{est}(t_{j\Delta_l}) \notin I_{id}\right) \geq \frac{1}{4^{\Delta_l}}$ so that $\mathbb{P}(\mathcal{S}_j) \geq \frac{1}{4^{\Delta_l}}$ and $\mathbb{E}[Y] \leq 4^{\Delta_l}$. Since Y denotes the index of the first phase during which Bob gets out of I_{id}, then $M' \in [\![(Y-1)\Delta_l; Y\Delta_l]\!]$ and in particular, $M' \leq Y\Delta_l$, so $\mathbb{E}[M'] \leq \Delta_l \times 4^{\Delta_l}$. By Proposition 1, we finally obtain $\mathbb{E}[M] \leq \Delta_l \times 4^{\Delta_l}$, what achieves the proof of Theorem 5.

□

4 Random Walk in 2D and 3D

As seen in Section 2, in a n-D space space, the movement of a player consists in following a random walk on a n-D grid. If at instant t, a player is at position $p = (p_1, p_2, \ldots, p_n)$, then they have $2n$ neighbors: $(p_1 - 1, p_2, \ldots, p_n)$, $(p_1 + 1, p_2, \ldots, p_n)$, $(p_1, p_2 - 1, \ldots, p_n)$, $\ldots$, $(p_1, p_2, \ldots, p_n - 1)$, $(p_1, p_2, \ldots, p_n + 1)$. The movement consists, at each integer instant, to chose one of those neighbors, each with probability $\frac{1}{2n}$.

In 2D, for example, this means that, at each instant, a moving player adds one of the following to his/her position: $(-1,0)$, $(1,0)$, $(0,-1)$, or $(0,1)$.

For our analysis, we will use the L^1 distance (Manhattan distance), that is, for two positions $p = (p_1, p_2)$ and $p' = (p'_1, p'_2)$, the distance is $d(p, p') = |p_1 - p'_1| + |p_2 - p'_2|$.

Let us call n the number of dimensions, supposed less than or equal to three. Let us prove that in a n-D space, we have a similar bound than in the 1D case.

Theorem 6. *In a n-D Euclidian space, with $n \leq 3$, and $\Delta_r = \left\lceil \frac{\log(1+\varepsilon)-\log(1-\varepsilon)}{\log(1+\frac{\varepsilon}{2})} \right\rceil$, and with two players moving, we have $\mathbb{E}[M] \leq \Delta_r \times \left(2^{n+1}\right)^{\Delta_r}$*

Let us assume, without loss of generality, that Bob is the player that triggers the $(i+1)$-th message, at instant t_{i+1}.

Let us call $\mathcal{B}_A(t)$ (resp. $\mathcal{B}_B(t)$) the L^1-ball of radius $\left\lceil d_{est}(t)\frac{\varepsilon}{2} \right\rceil$, and of center $\widetilde{p_A}(t)$ (resp. $\widetilde{p_B}(t)$). Thus, $\mathcal{B}_A(t)$ is the set of positions that are at a distance from $\widetilde{p_A}(t)$ less than or equal to $\left\lceil d_{est}(t)\frac{\varepsilon}{2} \right\rceil$ (this is the lower square on Figure 2a).

Remark 1. With $\mathcal{A}_{lc}$, the $(i+1)$-th message is sent when Bob is on the border of $\mathcal{B}_B(t_i)$.

Proof. With $\mathcal{A}_{lc}$, the $(i+1)$-th message is sent when Bob gets at a position that is at a distance at least $\frac{\varepsilon}{2}d_{est}(t_i)$ from $\widetilde{p_B}(t_i)$. As movement is on integer positions, the first positions satisfying this are all on the border of $\mathcal{B}_B(t_i)$. □

This ball $\mathcal{B}_B$ has 2^n faces of dimension $(n-1)$. We may draw cones over each of these faces, with $\widetilde{p_B}(t_i)$ as the apex: all points of the space will be in only one of the cones, except for points on the borders (see Figure 2b for a two-dimensional example, where the borders of the cones are the dashed lines). Let us call $\mathcal{R}$ the face that is included in the cone (or one of the cones) opposing the one containing $\widetilde{p_A}(t_i)$.

Before being able to identify the effect a message has on the estimated distance (Lemma 6), we analyse how far Bob's estimated position can get from Alice (Lemma 4).

Lemma 4. *If $\widetilde{p_B}(t_i) \neq \widetilde{p_A}(t_i)$, $\mathbb{P}\left(d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1})) \geq \left\lceil d_{est}(t_i)\left(1+\frac{\varepsilon}{2}\right)\right\rceil\right) \geq \frac{1}{2^n}$*

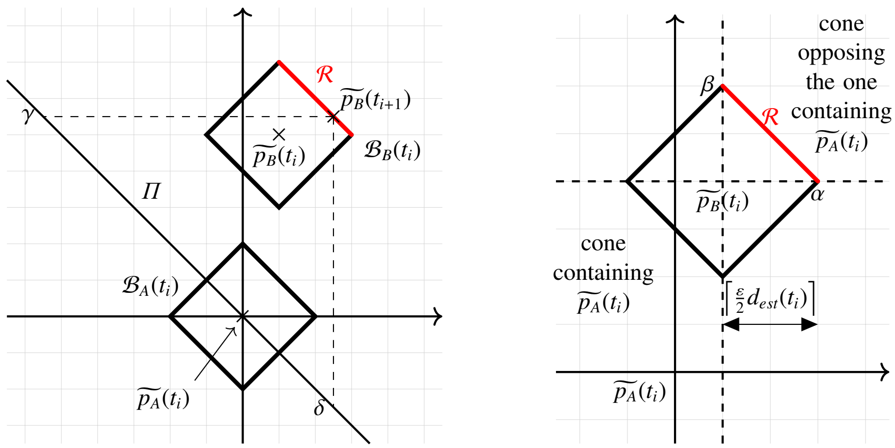
Proof. All points of $\mathcal{R}$ are at distance $\left\lceil d_{est}(t_i)\left(1+\frac{\varepsilon}{2}\right)\right\rceil$ of $\widetilde{p_A}(t_i)$ (for this, consider one of the endpoints of the face, like α on Figure 2b, for which all coordinates are the same as for $\widetilde{p_B}(t_i)$, except one, where the absolute value is larger by $\left\lceil \frac{\varepsilon}{2}d_{est}(t_i) \right\rceil$). Finally, as the random walk is symmetric, and by Remark 1, we have a probability of at least $\frac{1}{2^n}$ that Bob sends the $(i+1)$-th message by going on face $\mathcal{R}$. □

In Lemma 4, the movement of Alice is not taken into account. Let us call Π the hyperplane parallel to $\mathcal{R}$ and containing $\widetilde{p_A}(t_i)$ (see Figure 2a for a two-dimensional example).

Remark 2. As Π contains $\widetilde{p_A}(t_i)$, the center of $\mathcal{B}_A$, Π divides $\mathcal{B}_A$ into two halves of the same size.

Lemma 5. *At least half of the points p of $\mathcal{B}_A$ satisfy :*

$$d(p, \widetilde{p_B}(t_{i+1})) \geq d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1})).$$



(a) When Bob gets on $\mathcal{R}$, half of the possible positions of Alice are further away.

(b) One of the face of $\mathcal{B}_B$ is always sufficiently far away from $\widetilde{p_A}(t_i)$.

Fig. 2: Random walk, two-dimensional situation

Proof. By definition of the L^1-norm, and because Π is parallel to $\mathcal{R}$, if we draw, on Π, a polygon connecting n points that are the projections of $\widetilde{p_B}(t_{i+1})$ parallel to the n axes (γ and δ on Figure 2a), then all points of Π inside this polygon (including the borders) are all at the same distance to $\widetilde{p_B}(t_{i+1})$.

Also, by definition of $\mathcal{R}$, $\widetilde{p_A}(t_i)$ is inside the polygon. Thus, all points of the polygon are at a distance to $\widetilde{p_B}(t_{i+1})$ equal to $d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1}))$.

If we draw the L^1-ball of center $\widetilde{p_B}(t_{i+1})$ and of radius $d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1}))$, then the polygon is one of the faces of the ball. By Remark 2, we have that at least half of the points from $\mathcal{B}_A$ are outside this ball, with a distance to $\widetilde{p_B}(t_{i+1})$ higher than the radius of the ball. □

We can now look at the estimated distance.

Lemma 6. *As long as* $\widetilde{p_B}(t_i) \neq \widetilde{p_A}(t_i)$, $\mathbb{P}\left(d_{est}(t_{i+1}) \geq \left\lceil d_{est}(t_i)\left(1+\frac{\varepsilon}{2}\right)\right\rceil\right) \geq \frac{1}{2^{n+1}}$

Proof. As Alice does not get out of $\mathcal{B}_A$, we know that $\widetilde{p_A}(t_{i+1}) \in \mathcal{B}_A$. By Lemma 5, and by symmetry of the random movement, $d(\widetilde{p_A}(t_{i+1}), \widetilde{p_B}(t_{i+1})) \geq d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1}))$ with probability $\frac{1}{2}$. Thus, the result is the same as for Lemma 4, but with half as much probability. □

As we consider r-jumps, we have to adapt Lemma 1 as follows.

Lemma 7. *For all* $x \in I_{id}$, $r^{\Delta_r}(x) \geq d_0(1+\varepsilon)$.

Proof. $x \in I_{id} \Rightarrow x \geq d_0(1-\varepsilon) \Rightarrow r^{\Delta_r}(x) \geq r^{\Delta_r}(d_0(1-\varepsilon))$ since r is increasing, implying that $r^{\Delta_r}(x) \geq d_0(1-\varepsilon)\left(1+\frac{\varepsilon}{2}\right)^{\Delta_r}$ since $\forall x$, $r(x) \geq x\left(1+\frac{\varepsilon}{2}\right)$. Moreover, since, $\Delta_r \geq \frac{\log(1+\varepsilon)-\log(1-\varepsilon)}{\log\left(1+\frac{\varepsilon}{2}\right)}$, then $(1-\varepsilon)\left(1+\frac{\varepsilon}{2}\right)^{\Delta_l} \geq (1+\varepsilon)$ and $x \in I_{id} \Rightarrow r^{\Delta_r}(x) \geq d_0(1+\varepsilon)$ □

We are now ready to prove Theorem 6:

Proof. This proof is very similar to Theorem 2. By Lemma 6, we know that the probability of having a r-jump (as defined in Equation 4) at an instant t_i, is at least $\frac{1}{2^{n+1}}$.

With *phases* of length Δ_r and j the index of the phase containing jumps from $m_{(j-1)\Delta_r}$ to $m_{j\Delta_r-1}$, we have (i) $\mathcal{S}_j$: there is at least one $i \in [\![(j-1)\Delta_r; j\Delta_r]\!]$ such that $d_{est}(t_i) \notin I_{id}$ (ii) $\mathcal{S}'_j$: the phase j is composed only of r-jumps. (iii) $X_j = 1$ if $\mathcal{S}_j$ is true, 0 otherwise (iv) $X'_j = 1$ if $\mathcal{S}'_j$ is true, 0 otherwise (v) $Y = j$ if $X_j = 1$ and $X_k = 0$ for every $k < j$ and (vi) $Y' = j$ if $X'_j = 1$ and $X'_k = 0$ for every $k < j$. Thus, Y denotes the index of the first phase during which Bob gets out of I_{id}.

If $\mathcal{S}'_j$ is true, then $d_{est}(t_{j\Delta_r}) = r^{\Delta_r}(d_{est}(t_{(j-1)\Delta_r}))$. Thus, by Lemma 7, $\mathcal{S}'_j \Rightarrow \mathcal{S}_j$, so that $X'_j = 1 \Rightarrow X_j = 1$.

$$\text{Therefore } Y' = j \Rightarrow X'_j = 1 \Rightarrow X_j = 1 \Rightarrow Y \leq j \text{ and finally } \mathbb{E}[Y] \leq \mathbb{E}[Y'] \quad (8)$$

Moreover, we know that Y' follows a geometric distribution with parameter $\mathbb{P}(\mathcal{S}'_j) \geq \frac{1}{(2^{n+1})^{\Delta_r}}$ (because each jump has at least probability $\frac{1}{2^{n+1}}$ of being r), and $\mathbb{E}[Y'] \leq \left(2^{n+1}\right)^{\Delta_r}$. Thus, by Equation 8, we have $\mathbb{E}[Y] \leq \left(2^{n+1}\right)^{\Delta_r}$. Since Y denotes the index of the first phase during which Bob gets out of I_{id}, $M' \in [\![(Y-1)\Delta_r; Y\Delta_r]\!]$. In particular, $M' \leq Y\Delta_r$ and $\mathbb{E}[M'] \leq \Delta_r \times \left(2^{n+1}\right)^{\Delta_r}$. Finally, Proposition 1 proves that $\mathbb{E}[M] \leq \Delta_r \times \left(2^{n+1}\right)^{\Delta_r}$. □

Remark 3. If only one player moves, then $\mathbb{E}[M] \leq \Delta_r \times (2^n)^{\Delta_r}$

Proof. The proof is the same as for Theorem 6, noticing that $d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1})) = d_{est}(t_{i+1})$. □

5 Continuous Movement, Discrete Time

In this section, we present bounds on M for continuous movements.

5.1 1D Case

As we have seen in Section 2, in one dimension, the movement simply consists in adding to the position a random number following a uniform distribution on $[-1, 1]$.

The problem is that when player X gets out of $\mathcal{B}_X(t_i)$, then the next position may take several values: for example, if X got out by the left, then $\widetilde{p_X}(t_i)$ may take any value smaller than the left bound of $\mathcal{B}_X(t_i)$ and greater to this bound minus one (the biggest movement he may have done at the last instant before getting out).

Nevertheless, the equivalent of Theorem 5 still holds true in this setting:

Theorem 7. *If* $\varepsilon \in]0; 1[$ *, then with two players following a continuous random movement on* $\mathbb{R}$*,* $\mathbb{E}[M] \leq \Delta_l \times 4^{\Delta_l}$

To see this, we will again call m_i the transformation between $d_{est}(t_i)$ and $d_{est}(t_{i+1})$. This time we will say:

$$m_i \in \begin{cases} \mathscr{L} \text{ if } d_{est}(t_{i+1}) \leq d_{est}(t_i)\left(1 - \frac{\varepsilon}{2}\right) \\ \mathscr{R} \text{ if } d_{est}(t_{i+1}) \geq d_{est}(t_i)\left(1 + \frac{\varepsilon}{2}\right) \end{cases} \tag{9}$$

By definition of $\mathcal{A}_{lc}$, m_i has to be in either $\mathscr{L}$ or $\mathscr{R}$, and the probability is actually $\frac{1}{2}$ for both cases.

Using this, we have a result comparable to Lemma 1:

Lemma 8. *For all $x \in I_{id}$, if $j - i \geq \Delta_l$, and all $m_k \in \mathscr{L}$ for $k \in [\![i, j-1]\!]$ then $m_{j-1} \circ m_{j-2} \circ \cdots \circ m_i(x) \leq d_0(1 - \varepsilon)$.*

Proof. Let us call $m_i^{\Delta_l} = m_{j-1} \circ m_{j-2} \circ \cdots \circ m_i$.

Let us assume that $m_k \in \mathscr{L}$. All the m_k are increasing, as m_k is necessary of the form $x \mapsto x\left(1 - \frac{\varepsilon}{2}\right) - a_k$, with $a_k \in [0, 1]$. Thus, we have $x \in I_{id} \Rightarrow x \leq d_0(1+\varepsilon) \Rightarrow m_i^{\Delta_l}(x) \leq m_i^{\Delta_l}(d_0(1+\varepsilon))$, what implies that $m_i^{\Delta_l}(x) \leq d_0(1+\varepsilon)\left(1 - \frac{\varepsilon}{2}\right)^{\Delta_l}$, since $\forall k \in [\![i, j-1]\!]$ and $\forall x$, $m_k(x) \leq x\left(1 - \frac{\varepsilon}{2}\right)$. Moreover, since, $\Delta_l \geq \frac{\log(1-\varepsilon) - \log(1+\varepsilon)}{\log\left(1 - \frac{\varepsilon}{2}\right)}$ and $\log\left(1 - \frac{\varepsilon}{2}\right) < 0$, then $(1+\varepsilon)\left(1 - \frac{\varepsilon}{2}\right)^{\Delta_l} \leq (1 - \varepsilon)$ and $x \in I_{id} \Rightarrow m_i^{\Delta_l}(x) \leq d_0(1-\varepsilon)$ □

The proof of Theorem 7 is then a direct translation of the proof of Theorem 5.

5.2 2D Case

As we have seen in Section 2, in two dimensions, the movement consists in choosing an angle θ between 0 and 2π, and moving a distance ρ between 0 and 1 in that direction. Thus, at each instant k, a moving player X chooses θ_k and ρ_k following continuous distributions respectively on $[0, 2\pi]$ and $[0, 1]$, so that $p_X(k+1) = p_X(k) + (\rho_k, \theta_k)$, where (ρ_k, θ_k) is the vector with polar coordinates ρ_k and θ_k.

Our result is as follows:

Theorem 8. *With $\Gamma = 2\,\frac{\log(1+\varepsilon) - \log(1-\varepsilon)}{\log\left(1 + \frac{\varepsilon}{\sqrt{2}} + \frac{\varepsilon^2}{4}\right)}$, with two players following a random continuous movement in two dimensions as previously defined, and implementing $\mathcal{A}_{lc}$ we have: $\mathbb{E}[M] \leq \Gamma \times 8^{\Gamma}$.*

This time again, we will call Bob the player who gets out the first of his set of authorized positions with $\mathcal{A}_{lc}$, meaning that Bob is the player to initiate communication at instant t_{i+1}.

In this setting, we will use the euclidian distance: $\mathcal{B}_B(t_i)$ takes the form of a circle of center $\widetilde{p_B}(t_i)$ and of radius $\frac{\varepsilon}{2} d_{est}$. We will use the same general principle as before, considering only jumps of a single type. Let us call $r_{cm} : x \mapsto x\sqrt{\left(1 + \frac{\varepsilon^2}{4} + \frac{\varepsilon}{\sqrt{2}}\right)}$.

In order to identify r_{cm}-jumps, we will look at the annulus of inner circle $\mathcal{B}_B(t_i)$, and with an outer circle of radius $\frac{\varepsilon}{2} d_{est} + 1$. We will call $\mathcal{R}$ the portion of this annulus on the opposite side of $\widetilde{p_A}(t_i)$, (represented as a red hatched zone on Figure 3), that deviates not more than $\frac{\pi}{4}$ from the straight line between $\widetilde{p_A}(t_i)$ and $\widetilde{p_B}(t_i)$. More formally, with t the intersection between $\mathcal{B}_B$ and the line $(\widetilde{p_A}(t_i)\widetilde{p_B}(t_i))$, on the opposite side of $\widetilde{p_A}(t_i)$, then $\mathcal{R} = \left\{s, \widehat{s\widetilde{p_B}(t_i)t} \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right] \text{ and } d(s, \widetilde{p_B}(t_i)) \in \left[\frac{\varepsilon}{2}d_{est}(t_i), \frac{\varepsilon}{2}d_{est}(t_i) + 1\right]\right\}$.

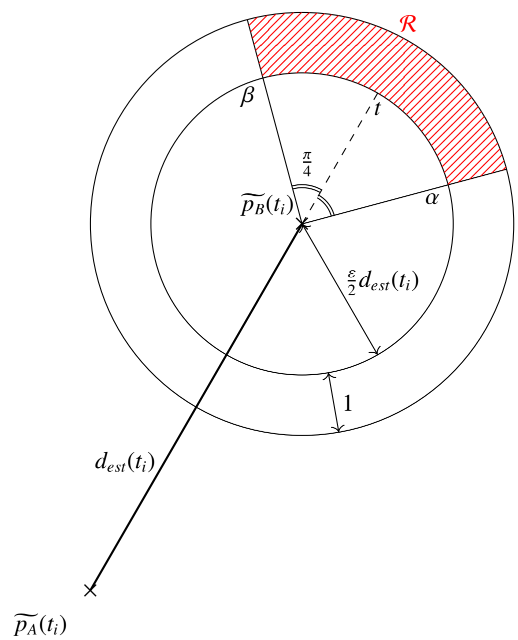


Fig. 3: Representation of the points corresponding to an r_{cm}-jump

We first identify the probability for Bob to send it position when getting in $\mathcal{R}$ in Lemma 9.

Lemma 9. *In two dimensions,* $\mathbb{P}(\widetilde{p_B}(t_{i+1}) \in \mathcal{R}) = \frac{1}{4}$.

Proof. As a player does not move more than one distance unit per time unit, the first instant where player Bob is outside of $\mathcal{B}_B(t_i)$, they will be in the annulus. Thus $\widetilde{p_B}(t_{i+1})$ is inside the annulus.

Without loss of generality, let us consider only the movement between Bob's initial position ($p_B(0)$, actually equal to $\widetilde{p_B}(0)$) and the position at time of the first message ($p_B(t_1)$, actually equal to $\widetilde{p_B}(t_1)$). Let us call $T = (p_0, p_1, \ldots, p_{t_1})$ the trajectory taken by player Bob to get on $\widetilde{p_B}(t_1)$, with p_t the position Bob had at instant t, where $t \in [\![0, t_1]\!]$. We have $p_0 = p_B(0)$, $p_1 = p_B(1)$, etc., and $p_{t_1} = \widetilde{p_B}(t_1)$.

See Figure 4 for a representation of the values.

Let us consider following random variables:

- R, taking the value of $d(p_0, p_{t_i})$.

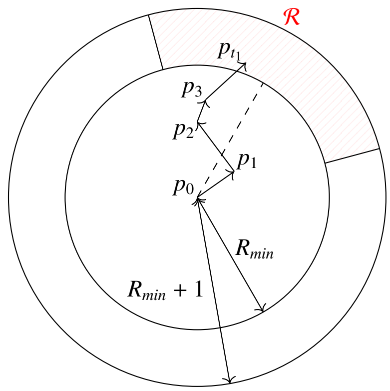


Fig. 4: Representation of the values used in proof of Lemma 9

- Θ, taking the value of the angle between the dashed line of Figure 4 and the position p_{t_i}.

Let us consider following event:

- T_k: the trajectory T is of length k.

As the random walk consists in randomly picking an angle θ_t and a distance ρ_t at every instant t, we have that $p_{t_1} = p_0 + (\rho_0, \theta_0) + (\rho_1, \theta_1) + \cdots + (\rho_{t_1-1}, \theta_{t_1-1})$, where (ρ_t, θ_t) is the vector of radius ρ_t and angle θ_t in polar coordinates.

Let us consider that T_k is true. Because the θ_t all follow a uniform distribution, the probability that $p_{t_1} = p_0 + (\rho_0, \theta_0) + (\rho_1, \theta_1) + \cdots + (\rho_{t_1-1}, \theta_{t_1-1})$ is "the same" as the probability that $p_{t_1} = p_0 + (\rho_0, \theta_0 + \gamma) + (\rho_1, \theta_1 + \gamma) + \cdots + (\rho_{t_1-1}, \theta_{t_1-1} + \gamma)$. More exactly, $\mathbb{P}(a \leq \Theta \leq b) = \mathbb{P}(a + \gamma \leq \Theta \leq b + \gamma)$ for all γ (regardless of the value of R).

Additionally, $\int_0^{2\pi} f_\Theta(x)dx = 1$, leading to the fact that $\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} f_\Theta(x)dx = \frac{1}{4}$.

We also have $\int_{R_{min}}^{R_{min}+1} f_R(x)dx = 1$. Thus, $\int_{R_{min}}^{R_{min}+1} \int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} f_{\Theta,R}(x,y)dxdy = \frac{1}{4}$.

Moreover, we have $\int_{R_{min}}^{R_{min}+1} \int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} f_{\Theta,R}(x,y)dxdy = \mathbb{P}(\widetilde{p_B}(t_1) \in \mathcal{R})$. As we supposed that T_k is true, we have that $\mathbb{P}(\widetilde{p_B}(t_1) \in \mathcal{R} \mid t_1 = k) = \frac{1}{4}$

By the law of total probability, we have that $\sum_0^{+\infty} \Big(\mathbb{P}(\widetilde{p_B}(t_1) \in \mathcal{R} \mid t_1 = k) \times \mathbb{P}(t_i = k)\Big) = \mathbb{P}(\widetilde{p_B}(t_1)) \in \mathcal{R}) = \frac{1}{4}$.

This remains true if we replace instants $0, 1, 2, \ldots, t_1$ by $t_i, t_i + 1, t_i + 2, \ldots, t_{i+1}$, proving this lemma. □

We may then identify, in Lemma 10 and Lemma 11, situations where r_{cm} appears.

Lemma 10. *With two players moving in two dimensions,*

$$\mathbb{P}\Big(d_{est}(t_{i+1}) \geq r_{cm}(d_{est}(t_i)) \mid \widetilde{p_B}(t_{i+1}) \in \mathcal{R}\Big) \geq \frac{1}{2}.$$

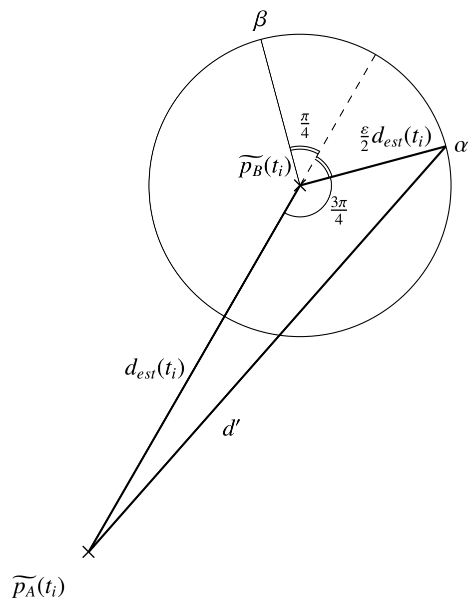


Fig. 5: Representation of the different values used to measure an r_{cm}-jump

Proof. Let us assume $\widetilde{p_B}(t_{i+1}) \in \mathcal{R}$.

The two points of $\mathcal{R}$ that are closest to $\widetilde{p_A}(t_i)$ are the rightmost and leftmost points that are both on $\mathcal{R}$ and the border of $\mathcal{B}_B(t_i)$ (α and β on Figure 3). Thus, if we call d' the distance between $\widetilde{p_A}(t_i)$ and α, we have $d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1})) \geq d'$.

As can be seen on Figure 5, the value of d' can be resolved by the law of cosines, relatively to the value of $d_{est}(t_i)$:

$$d' = \sqrt{d_{est}(t_i)^2 + \frac{\varepsilon^2}{4} d_{est}(t_i)^2 - d_{est}(t_i)^2 \varepsilon \cos\left(\frac{3\pi}{4}\right)} = d_{est}(t_i) \sqrt{\left(1 + \frac{\varepsilon^2}{4} + \frac{\varepsilon}{\sqrt{2}}\right)}$$

This corresponds to r_{cm}.

Thus, $\mathbb{P}(d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1})) \geq r_{cm}(d_{est}(t_i)) | \widetilde{p_B}(t_{i+1}) \in \mathcal{R}) = 1$.

We may then notice that, as player Alice remains inside $\mathcal{B}_A(t_i)$, the probability that $\widetilde{p_A}(t_{i+1})$ is further away from $\widetilde{p_B}(t_{i+1})$ than $\widetilde{p_A}(t_i)$ is at least one half. This gives us the final result. □

Lemma 11. *With two players moving,* $\mathbb{P}\Big(d_{est}(t_{i+1}) \geq r_{cm}(d_{est}(t_i))\Big) \geq \frac{1}{8}$

Proof. The proof of Lemma 11 is now immediate with Lemma 9, Lemma 10, and the law of total probability. □

The last needed property, is that successions of r_{cm} will make $\mathcal{A}_{id}$ send a message:

Lemma 12. *With* $\Gamma = \frac{\log(1+\varepsilon) - \log(1-\varepsilon)}{\log\left(\sqrt{1+\frac{\varepsilon^2}{4}+\frac{\varepsilon}{\sqrt{2}}}\right)}$, *for all* $x \in I_{id}$, ${r_{cm}}^{\Gamma}(x) \geq d_0(1+\varepsilon)$.

Proof. $x \in I_{id} \Rightarrow x \geq d_0(1-\varepsilon) \Rightarrow {r_{cm}}^{\Gamma}(x) \geq {r_{cm}}^{\Gamma}(d_0(1-\varepsilon))$ since r_{cm} is increasing, so that ${r_{cm}}^{\Gamma}(x) \geq d_0(1-\varepsilon)\left(\sqrt{\left(1+\frac{\varepsilon^2}{4}+\frac{\varepsilon}{\sqrt{2}}\right)}\right)^{\Gamma}$. Moreover, by definition of Γ, by applying the right exponential, $(1-\varepsilon)\left(\sqrt{\left(1+\frac{\varepsilon^2}{4}+\frac{\varepsilon}{\sqrt{2}}\right)}\right)^{\Gamma} \geq (1+\varepsilon)$, so that finally $x \in I_{id} \Rightarrow {r_{cm}}^{\Gamma}(x) \geq d_0(1+\varepsilon)$. □

Proof. We may now prove Theorem 8, with the same reasoning as Theorems 2, 3 and 6.

By Lemma 11, we know that the probability of having a jump that increases distance more than r_{cm} at an instant t_i, is at least $\frac{1}{8}$.

With *phases* of length Γ, and

1. $\mathcal{S}_j$: there is at least one $i \in [\![(j-1)\Gamma; j\Gamma]\!]$ such that $d_{est}(t_i) \notin I_{id}$
2. $\mathcal{S}'_j$: the phase j is composed only of jumps so that the distance increases more than with r_{cm}. That is, for all $i \in [\![(j-1)\Gamma; j\Gamma-1]\!]$, $d_{est}(t_{i+1}) \geq r_{cm}(d_{est}(t_i))$.
3. $X_j = 1$ if $\mathcal{S}_j$ is true, 0 otherwise
4. $X'_j = 1$ if $\mathcal{S}'_j$ is true, 0 otherwise
5. $Y = j$ if $X_j = 1$ and $X_k = 0$ for every $k < j$ and
6. $Y' = j$ if $X'_j = 1$ and $X'_k = 0$ for every $k < j$. Thus, Y denotes the index of the first phase during which Bob gets out of I_{id}.

We know that Y' follows a geometric distribution with parameter $\mathbb{P}(\mathcal{S}'_j) \geq \frac{1}{8^{\Gamma}}$ (because each jump has at least probability $\frac{1}{8}$ of complying to $\mathcal{S}'_j$), and $\mathbb{E}[Y'] \leq 8^{\Gamma}$. Thus, by Lemma 12, we have $\mathbb{E}[Y] \leq 8^{\Gamma}$. Since Y denotes the index of the first phase during which Bob gets out of I_{id}, $M' \in [\![(Y-1)\Gamma; Y\Gamma]\!]$. In particular, $M' \leq Y\Gamma$ and $\mathbb{E}[M'] \leq \Gamma \times 8^{\Gamma}$. Finally, Proposition 1 proves that $\mathbb{E}[M] \leq \Gamma \times 2^8\Gamma$. □

5.3 3D Case

Theorem 9. *With* $\Gamma = 2\,\frac{\log(1+\varepsilon)-\log(1-\varepsilon)}{\log\left(1+\frac{\varepsilon}{\sqrt{2}}+\frac{\varepsilon^2}{4}\right)}$*, with two players following a random continuous movement in 3D and implementing* $\mathcal{A}_{lc}$*, we have* $\mathbb{E}[M] \leq \Gamma \times 14^{\Gamma}$.

The reasoning is very similar to the two dimension case, the main difference being that $\mathcal{B}_B$ is now a sphere. Thus, $\mathcal{R}$ is now a portion of a spherical shell (instead of an annulus). The same definition of $\mathcal{R}$ as in the 2D case is still valid: with t the intersection between $\mathcal{B}_B$ and the line $(\widetilde{p_A}(t_i)\widetilde{p_B}(t_i))$, on the opposite side of $\widetilde{p_A}(t_i)$, then we have the definition $\mathcal{R} = \left\{s, s\widehat{\widetilde{p_B}(t_i)}t \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right] \text{ and } d(s, \widetilde{p_B}(t_i) \in \left[\frac{\varepsilon}{2}d_{est}(t_i), \frac{\varepsilon}{2}d_{est}(t_i)+1\right]\right\}$. It can also be seen as the previous $\mathcal{R}$ from Figure 3, but rotated with respect to the line $(\widetilde{p_A}(t_i)\widetilde{p_B}(t_i))$.

Lemma 13. *In the 3D case,* $\mathbb{P}(\widetilde{p_B}(t_{i+1}) \in \mathcal{R}) = \frac{2-\sqrt{2}}{4}$.

Proof. The first instant when player Bob gets outside of $\mathcal{B}_B(t_i)$, they will be in the spherical shell of inner sphere $\mathcal{B}_B(t_i)$, and with the outer sphere of same center, but with a radius longer of one distance unit ($\frac{\varepsilon}{2}d_{est}(t_i)+1$).

Let us use the spherical coordinates, centered on $\widetilde{p_B}(t_i)$, and with the z axis pointing towards t. The solid angle covered by $\mathcal{R}$ is the surface, on the unit sphere, of the zone where the colatitude is smaller than $\frac{\pi}{4}$: $\int_0^{\frac{\pi}{4}} \int_0^{2\pi} \sin(\phi) d\theta d\phi = 2\pi\left(1 - \frac{\sqrt{2}}{2}\right)$.

As the whole space is represented by 4π, this means that $\mathcal{R}$ takes $\left(\frac{1}{2} - \frac{\sqrt{2}}{4}\right) \approx 15\%$ of the spherical shell.

As movement is symmetric with respect to the center of the spherical shell, we have probability $\left(\frac{1}{2} - \frac{\sqrt{2}}{4}\right)$ that $\widetilde{p_B}(t_{i+1}) \in \mathcal{R}$. □

Lemma 14. *With two players moving in 3D,* $\mathbb{P}\left(d_{est}(t_{i+1}) \geq r_{cm}(d_{est}(t_i)) \mid \widetilde{p_B}(t_{i+1}) \in \mathcal{R}\right) \geq \frac{1}{2}$.

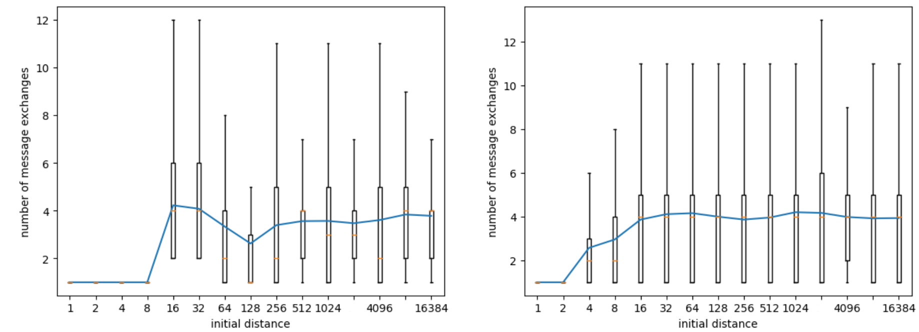
Proof. On all planes containing the line $(\widetilde{p_A}(t_i)\widetilde{p_B}(t_i))$, the points of $\mathcal{R}$ closest to $\widetilde{p_A}(t_i)$ follow the construction of α and β on Figure 5. Thus, as in the proof of Lemma 10, $\mathbb{P}\left(d(\widetilde{p_A}(t_i), \widetilde{p_B}(t_{i+1})) \geq r_{cm}(d_{est}(t_i)) \mid \widetilde{p_B}(t_{i+1}) \in \mathcal{R}\right) = 1$.

For any point x outside $\mathcal{B}_A(t_i)$, more than half of the points y inside $\mathcal{B}_A(t_i)$ satisfy $d(x, y) \geq d(x, \widetilde{p_A}(t_i))$. Thus, as $\widetilde{p_A}(t_{i+1})$ remains inside $\mathcal{B}_A(t_i)$, and $\widetilde{p_B}(t_{i+1}) \in \mathcal{R} \Rightarrow \widetilde{p_B}(t_{i+1}) \notin \mathcal{B}_A(t_i)$, we get our result. □

Proof. To prove Theorem 9, we first use Lemma 13 and Lemma 14: $\mathbb{P}\left(d_{est}(t_{i+1}) \geq r_{cm}(d_{est}(t_i))\right) \geq \frac{2-\sqrt{2}}{8}$. Thus, with phases of length Γ, the probability for a phase to contain only jumps that make distance higher than with r_{cm}-jumps is $\left(\frac{8}{2-\sqrt{2}}\right)^{\Gamma} \approx 14^{\Gamma}$. Finally, $\mathbb{E}[M] \leq \Gamma \times 14^{\Gamma}$. □

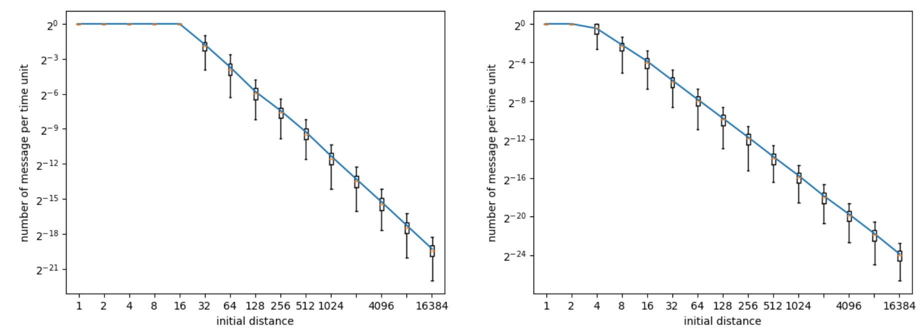
6 Experiments

In order to analyze in practice the performance of $\mathcal{A}_{lc}$, we propose simulation results. More precisely, we execute both $\mathcal{A}_{lc}$ and $\mathcal{A}_{id}$ with the same set of random movements (of one or two players) and we display M, the number of message exchanges induced by $\mathcal{A}_{lc}$ at the time the first message is induced by $\mathcal{A}_{id}$. We perform simulations for different values of the initial distance (d_0) and maximum error (ε) and for each set of parameters. Everywhere, we repeat the experiments 500 times to account for the stochastic nature of the movements. In all the plots, the blue lines indicate the average value, while the orange bars indicate the median values and the boxes indicate $Q1$, the first quartile and $Q3$, the third quartile. The lower whisker takes the values of the lowest reference point that is in the range $[Q1 - 1.5 \times IQR; Q1]$, where $IQR = Q3 - Q1$. Similarly, the upper whisker shows the highest reference point in the $[Q3; Q3 + 1.5 \times IQR]$ range. The results corresponding to the theoretical framework considered in Section 3 and Section 4 are presented in Section 6.1, while we present in Section 6.2 simulation results based on actual traces of games of Heroes of Newerth [1]. In particular, we use these traces to compare the behavior of $\mathcal{A}_{lc}$ with the behavior of solutions that are currently implemented in online games and that are based on fixed frequency messages.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.5$

Fig. 6: One player moving, random walk, 1D: M depending on initial distance



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.5$

Fig. 7: One player moving, random walk, 1D: messages per time unit

6.1 Synthetic Traces

The first set of simulations correspond to the setting of Section 3.1. In the 1D case, when only one player moves, the evolution of M with the initial distance is depicted in Figure 6a ($\varepsilon = 0.1$) and Figure 6b ($\varepsilon = 0.5$). As expected, we can observe that M remains bounded and does not depend much on the initial distance (except when the distance is very small with respect to movement amplitudes). Even though constants are smaller than those proved in Theorem 2 and Theorem 3, the results are as expected by the theoretical analysis. Figure 7a and Figure 7b depict the actual number of messages sent when using $\mathcal{A}_{lc}$, as a function of the initial distance for $\varepsilon = 0.1$ and $\varepsilon = 0.5$. We can observe that the number of messages generated by $\mathcal{A}_{lc}$ quadratically decreases with the distance between the players (slope -2 in log-log scale), which is a desirable property, since maintaining an approximate distance should be less expensive when player avatars are distant. We also plot the evolution of M with the given maximal tolerated error, ε in

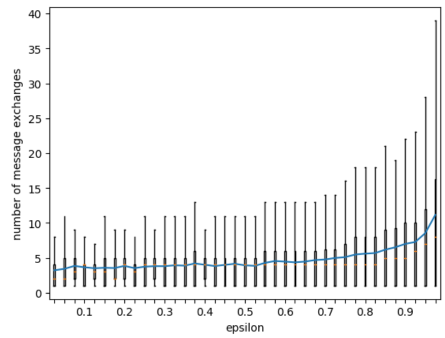
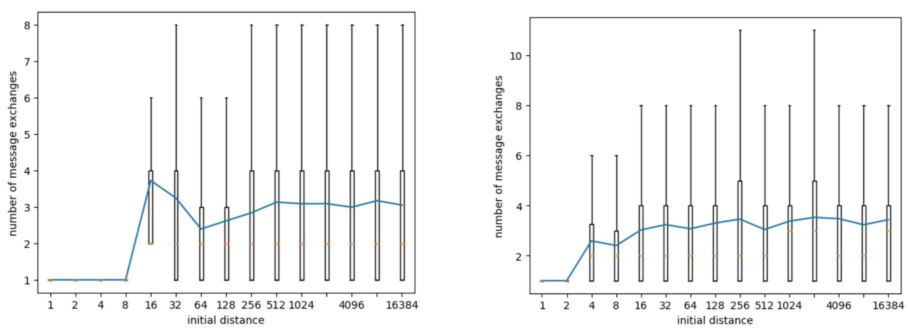


Fig. 8: One player moving, random walk, 1D: value of M depending on ε, for $d_0 = 400$

Figure 8. We can observe that M increases when ε gets close to 1, what suggests that the dependance on ε in our theoretical bounds is unavoidable.

We performed the same set of experiments when both players move in a 1D-space, 2D-space, or 3D-space, and obtained very similar results, which are not included here for the sake of conciseness, but can be found below.



(a) $\varepsilon = 0.1$

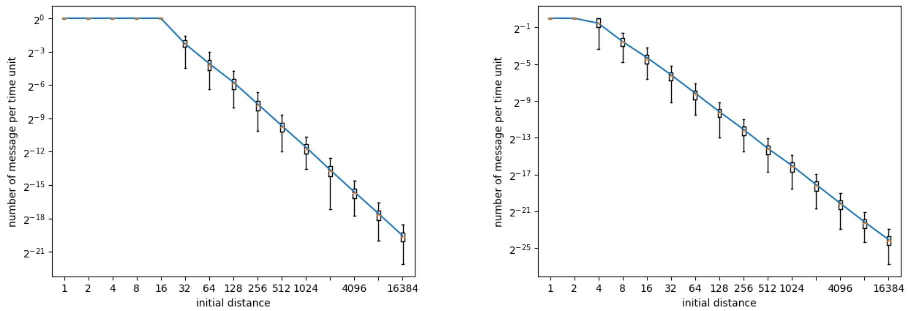
(b) $\varepsilon = 0.5$

Fig. 9: Two players moving, random walk, 1D: M depending on initial distance

When both players move, as can be seen on figures 9, 10, and 11, curves show similar behavior as when only one player moves.

Figures 12, 13, and 14 show results for two players moving in a 2D-space. In this case too, M remains bounded when the initial distance changes (from approximately 4 times more messages in 1D to approximately 6 times more messages in 2D). We also observe that the actual number of messages quadratically decreases with the initial distance, as in the 1D case.

In the 3D case, $\mathbb{E}[M]$ still does not depend on the initial distance, as can be seen on Figure 15a and Figure 15b. Surprisingly, its value appears no longer to depend on ε,



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.5$

Fig. 10: Two players moving, random walk, 1D: messages per time unit with $\mathcal{A}_{lc}$

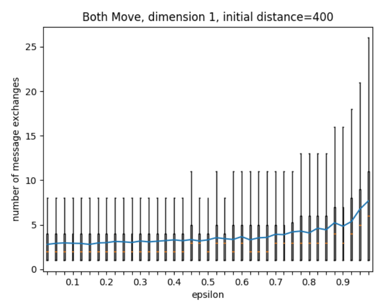


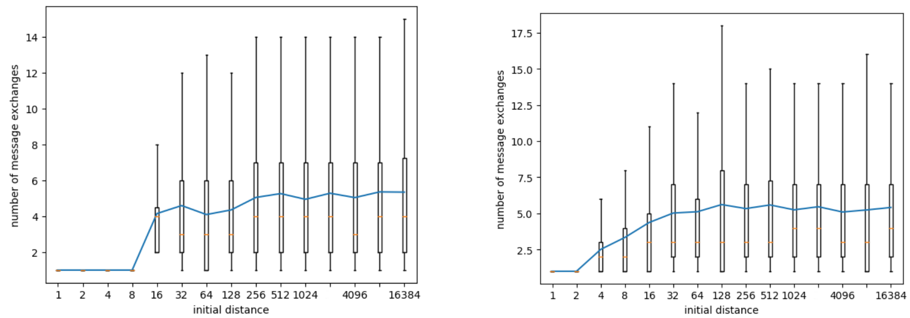
Fig. 11: Two players moving, random walk, 1D: M depending on ε, for $d_0 = 400$

Figure 17 happens to no longer grow when ε is close to 1. This may hint that it would be possible to get upper bounds on M that do not depend on ε ; but as ε is already a constant, the theoretical benefit would be small.

This qualitative analysis is exactly the same when considering continuous movement instead of a random walk, as can be seen on figures 19 through 21 (Appendix A): M has an upper bound, and depends on ε, except maybe for the 3D case, and the number of messages generated by $\mathcal{A}_{lc}$ decreases quadratically with the distance.

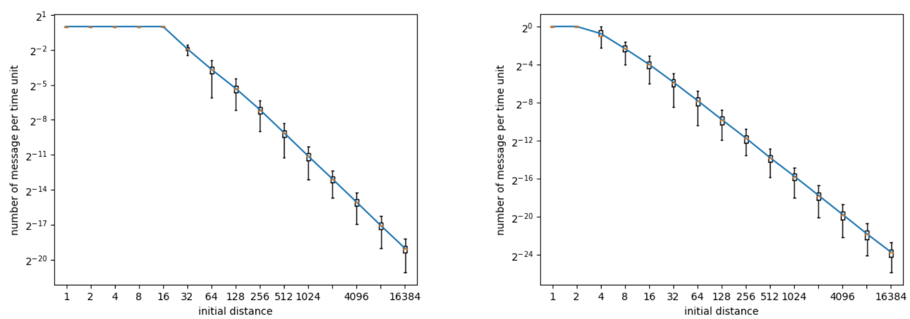
6.2 Actual Traces

Comparison of $\mathcal{A}_{lc}$ with fixed frequency strategies. In order to assess the performance of $\mathcal{A}_{lc}$, we finally compare it to the fixed frequency strategy that is used in practice in actual games [2], and denoted by $\mathcal{A}_{ff}$. This algorithm does not take a maximal error as parameter, but a fixed wait time w between message exchange of any pair of players. The traces provided in [8] contain time-stamped information on 98 games of Heroes of Newerth [1] and were used in [9] with the purpose of building mobility models. They contain the evolution of positions of 10 players in each trace. Therefore,



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.5$

Fig. 12: Two players moving, random walk, 2D: M depending on initial distance



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.5$

Fig. 13: Two players moving, random walk, 2D: messages per time unit with $\mathcal{A}_{lc}$

a wait time of w induces $\frac{9*10}{w}$ messages at each time step (on average). Even if a smaller w makes information more accurate, $\mathcal{A}_{ff}$ comes without guarantee on maximal error violations, contrarily to $\mathcal{A}_{lc}$. To evaluate the performance of $\mathcal{A}_{ff}$ in terms of accuracy, we simulated its behavior for several values of ε and w. We counted the *number of violations per time unit*, that is, the number of distance estimates among the players that violate Equation 2. As there are ten players, and each one has an estimate for all nine others, the number of violations has a maximum of 90 for one time unit. Figure 18 depicts the number of violations for different values of ε and w. We observe that the number of violations increases very quickly with w.

In order to perform a fair comparison between $\mathcal{A}_{lc}$ and $\mathcal{A}_{ff}$, we used the following protocol. First, we ran $\mathcal{A}_{lc}$ for several values of ε, and we measured the resulting average number of messages per time unit. Then, we plugged obtained value as w time in $\mathcal{A}_{ff}$, so that we can compare both algorithms in terms of accuracy (to estimate approximated distance) while they use exactly the same average message frequency. The average proportion of violations is shown in bold font in Table 1, along with the optimal

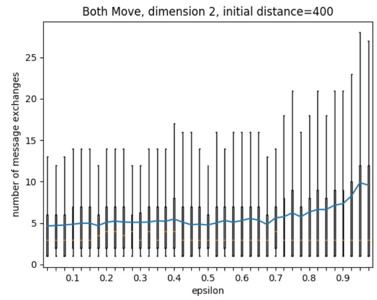
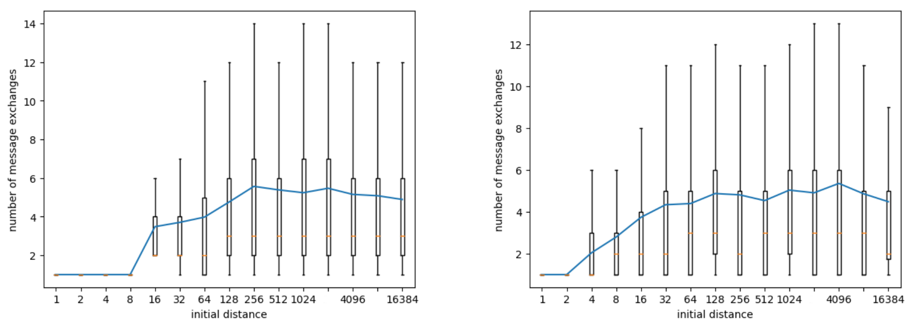


Fig. 14: Two players moving, random walk, 2D: M depending on ε, for $d_0 = 400$



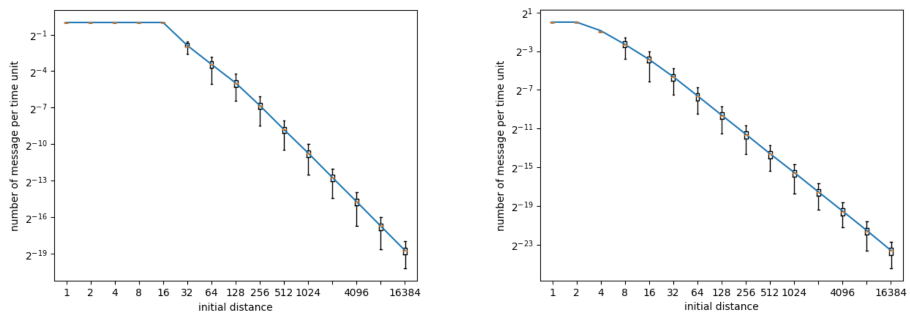
(a) $\varepsilon = 0.1$

(b) $\varepsilon = 0.5$

Fig. 15: Two players moving, random walk, 3D: M depending on initial distance

number of messages, that is, $\mathcal{A}_{id}$, for different values of ε. We can observe that $\mathcal{A}_{lc}$ is far better than $\mathcal{A}_{ff}$ for satisfying Equation 2. For instance, it sends only 10.44 messages per time unit for $\varepsilon = 0.1$. With $\mathcal{A}_{ff}$, the only way to ensure Equation 2 is by having $w = 1$. This would lead to 90 messages per time unit with $w = 1$, that is, about ten times more than $\mathcal{A}_{lc}$.

Influence of better prediction strategies. As mentioned in Section 1.1, Dead-reckoning is a popular method for reducing the error on positions of elements of an online game. This is why we wanted to see if adding Dead-reckoning adds to the benefits of our algorithm. To do this, we rely on a position prediction algorithm, which is based on the speed. Speed is calculated based on the two last known positions, and is used to extrapolate the previous known position. The results of the same experiment as above, with this prediction algorithm, are shown on Table 1, within parenthesis. We can observe that the number of message exchanged in $\mathcal{A}_{lc}$ decreases more significantly than $\mathcal{A}_{id}$. Moreover, Dead-reckoning seems to be more beneficial to $\mathcal{A}_{lc}$ than to $\mathcal{A}_{ff}$, as the decrease in message number is not compensated for in terms of violations by the improved prediction precision.



(a) for $\varepsilon = 0.1$

(b) $\varepsilon = 0.5$

Fig. 16: Two players moving, random walk, 3D: messages per time unit with $\mathcal{A}_{lc}$

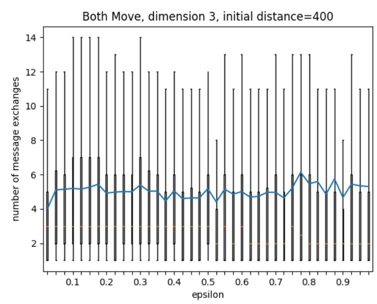


Fig. 17: Two players moving, random walk, 3D: M depending on ε, for $d_0 = 400$

7 Conclusion and future work

In this paper, we propose a distributed algorithm $\mathcal{A}_{lc}$, for each player to estimate the distance separating them from each other player, with a relative condition on the error. This type of property is desirable in DVE such as online games. We prove that (in a restricted setting), this algorithm is optimal in terms of number of message exchanges up a to a constant factor. We also show through simulations, based on actual game traces, that $\mathcal{A}_{lc}$ performs significantly less communications than the fixed frequency algorithm which is commonly used in online game, while bounding the error.

A summary of our bounds can be found in the following table:

	random walk	continuous movement
1D case	$\min\left(\Delta_l \times 2^{\Delta_l} \,;\, \left\lceil \frac{4}{\pi}{\Delta_l}^2 \right\rceil \times 8\right)$	$\Delta_l \times 4^{\Delta_l}$
2D case	$\Delta_r \times 8^{\Delta_r}$	$\Gamma \times 8^{\Gamma}$
3D case	$\Delta_r \times 16^{\Delta_r}$	$\Gamma \times 14^{\Gamma}$

This work opens several perspectives. The first one is to extend the theoretical results proved in this paper, either by improving the constants or by increasing the scope

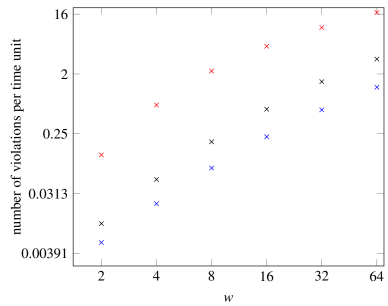


Fig. 18: Number of violations with $\mathcal{A}_{ff}$, depending on time to wait between two messages, with $\varepsilon = 0.1$ (red), $\varepsilon = 0.5$ (black), and $\varepsilon = 0.9$ (blue), on log-log scale

Table 1: Comparison of $\mathcal{A}_{lc}$ and $\mathcal{A}_{ff}$, **without Dead-reckoning** (with Dead-reckoning)

	$\mathcal{A}_{id}$	$\mathcal{A}_{lc}$		$\mathcal{A}_{ff}$		
ε	msg/time unit	messages per time unit	violations	w	msg/time unit	violations
0.1	**3.26** (2.23)	**10.44** (4.71)	0.0	**9** (19)	**10.00** (4.73)	**2.9%** (5.13%)
0.2	**1.49** (1.24)	**5.41** (3.02)	0.0	**17** (30)	**5.30** (3.00)	**2.74%** (4.66%)
0.3	**0.91** (0.84)	**3.60** (2.26)	0.0	**25** (40)	**3.60** (2.25)	**2.6%** (4.26%)
0.4	**0.63** (0.62)	**2.65** (1.81)	0.0	**34** (50)	**2.65** (1.80)	**2.53%** (3.88%)
0.5	**0.46** (0.46)	**2.07** (1.50)	0.0	**43** (60)	**2.09** (1.50)	**2.42%** (3.51%)

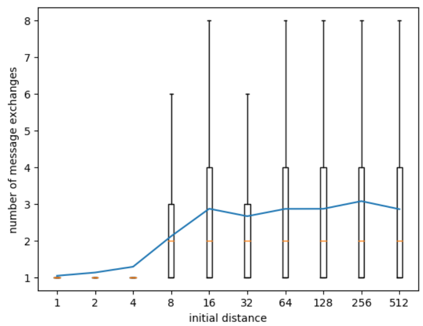
of the results and to consider more sophisticated prediction algorithms. Another longer term perspective is to extend the set of properties that can be maintained in DVEs at the price of re-computations and a (constant) increase in exchanged messages. It was known in the literature that maintaining the positions was possible with no increase in the number of messages and the present paper shows that a constant increase is enough to maintain relative distances. Extending the class of such properties is highly desirable, both in theory and practice.

References

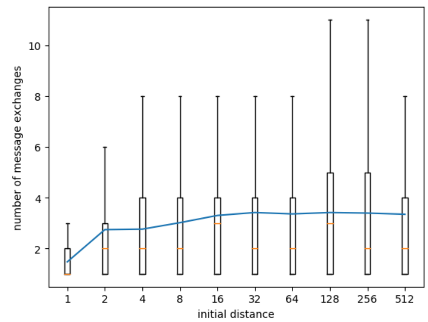
1. Heroes of Newerth - Home, http://www.heroesofnewerth.com/, accessed 27 August 2019
2. Source Multiplayer Networking - Valve Developer Community, https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking, accessed 27 August 2019
3. IEEE standard for distributed interactive simulation–application protocols. IEEE Std 1278.1-2012 (Revision of IEEE Std 1278.1-1995) pp. 1–747 (Dec 2012). https://doi.org/10.1109/IEEESTD.2012.6387564
4. Aggarwal, S., Banavar, H., Khandelwal, A., Mukherjee, S., Rangarajan, S.: Accuracy in dead-reckoning based distributed multi-player games. In: Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games. pp. 161–165. NetGames '04, ACM, New York, NY, USA (2004)

5. Bharambe, A., Douceur, J., Lorch, J., Moscibroda, T., Pang, J., Seshan, S., Zhuang, X.: Donnybrook: Enabling large-scale, high-speed, peer-to-peer games. In: SIGCOMM 2008: ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Seattle, WA. pp. 389–400. Association for Computing Machinery, Inc. (August 2008)
6. Boulanger, J.S., Kienzle, J., Verbrugge, C.: Comparing interest management algorithms for massively multiplayer games. In: Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games. NetGames '06, ACM, New York, NY, USA (2006)
7. Cai, W., Lee, F.B.S., Chen, L.: An auto-adaptive dead reckoning algorithm for distributed interactive simulation. In: Workshop on Parallel and Distributed Simulation (1999)
8. Carlini, E., Lulli, A.: Data used for submission entitled "A spatial analysis of Multiplayer Online Battle Arena mobility traces" (May 2017). https://doi.org/10.5281/zenodo.583600, https://doi.org/10.5281/zenodo.583600
9. Carlini, E., Lulli, A.: A spatial analysis of multiplayer online battle arena mobility traces. In: Euro-Par 2017: Parallel Processing Workshops. pp. 496–506. Springer International Publishing, Cham (2018)
10. Fullford, D.: Distributed interactive simulation: it's past, present, and future. In: Proceedings Winter Simulation Conference. pp. 179–185 (Dec 1996)
11. Kharitonov, V.Y.: Motion-aware adaptive dead reckoning algorithm for collaborative virtual environments. In: Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry. pp. 255–261. VRCAI '12, ACM, New York, NY, USA (2012)
12. Li, Z., Tang, X., Cai, W., Li, X.: Compensatory dead-reckoning-based update scheduling for distributed virtual environments. SIMULATION **89**(10), 1272–1287 (2013)
13. Liu, E.S., Theodoropoulos, G.K.: Interest management for distributed virtual environments: A survey. ACM Comput. Surv. **46**(4), 51:1–51:42 (Mar 2014)
14. Marshall, D., Mcloone, S., Ward, T., Delaney, D.: Does reducing packet transmission rates help to improve consistency within distributed interactive applications? (01 2006)
15. Mauve, M., Vogel, J., Hilt, V., Effelsberg, W.: Local-lag and timewarp: Providing consistency for replicated continuous applications. Multimedia, IEEE Transactions on **6**, 47 – 57 (03 2004)
16. Millar, J.R., Hodson, D.D., Peterson, G.L., Ahner, D.K.: Consistency and fairness in real-time distributed virtual environments: Paradigms and relationships. Journal of Simulation **11**(3), 295–302 (Aug 2017)
17. Ricci, L., Carlini, E.: Distributed virtual environments: From client server to cloud and P2P architectures. In: 2012 International Conference on High Performance Computing Simulation (HPCS). pp. 8–17 (July 2012)
18. Zhou, S., Cai, W., Lee, B.S., Turner, S.J.: Time-space consistency in large-scale distributed virtual environments. ACM Trans. Model. Comput. Simul. **14**(1), 31–47 (Jan 2004)

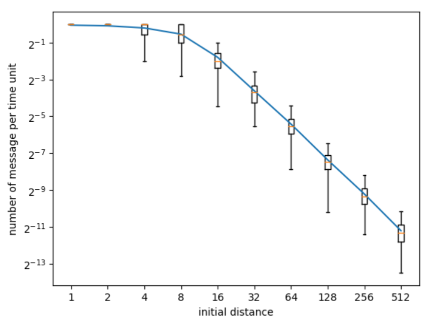
A Figures for Continuous Movement



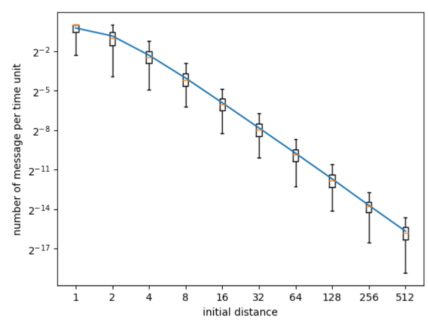
(a) M depending on initial distance, for $\varepsilon = 0.1$



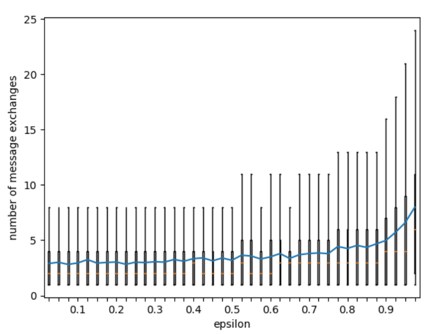
(b) M depending on initial distance, for $\varepsilon = 0.5$



(c) Messages per time unit with $\mathcal{A}_{lc}$, for $\varepsilon = 0.1$

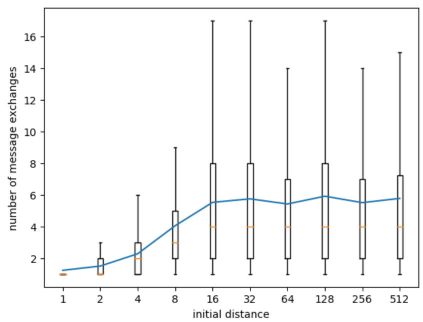


(d) Messages per time unit with $\mathcal{A}_{lc}$, for $\varepsilon = 0.5$

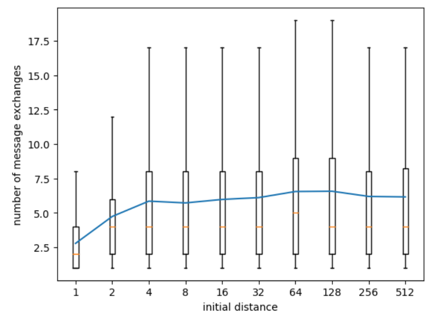


(e) M depending on ε, for $d_0 = 400$

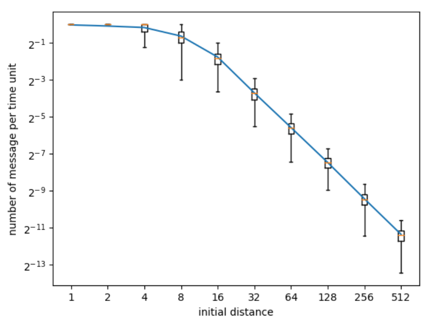
Fig. 19: Values in the 1D case when both players follow a continuous movement



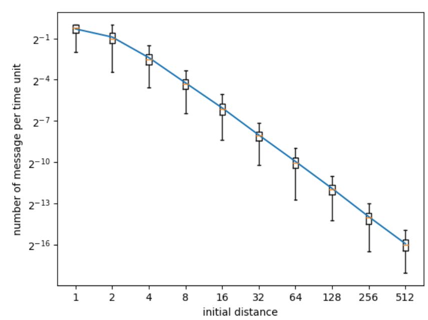
(a) M depending on initial distance, for $\varepsilon = 0.1$



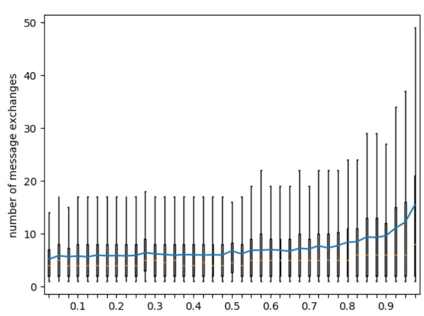
(b) M depending on initial distance, for $\varepsilon = 0.5$



(c) Messages per time unit with $\mathcal{A}_{lc}$, for $\varepsilon = 0.1$

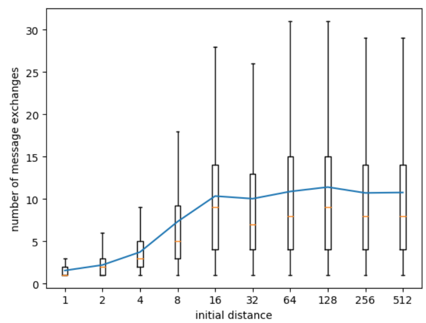


(d) Messages per time unit with $\mathcal{A}_{lc}$, for $\varepsilon = 0.5$

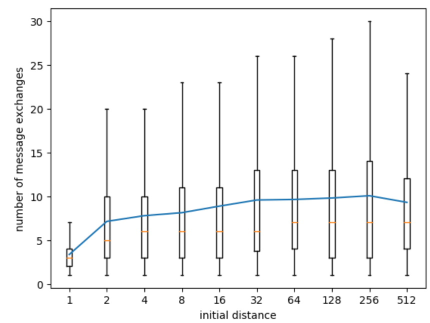


(e) M depending on ε, for $d_0 = 400$

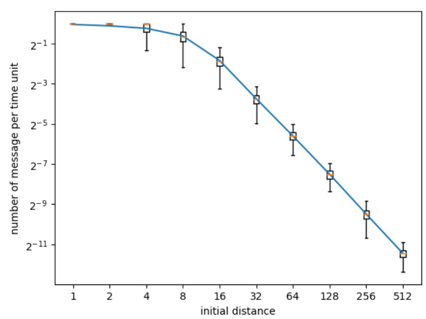
Fig. 20: Values in the 2D case when both players follow a continuous movement



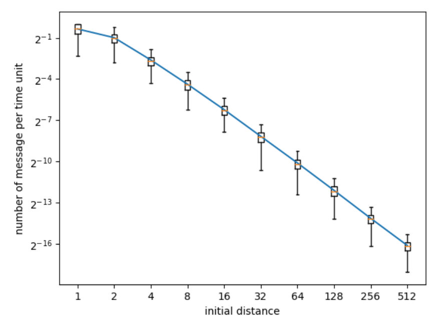
(a) M depending on initial distance, for $\varepsilon = 0.1$



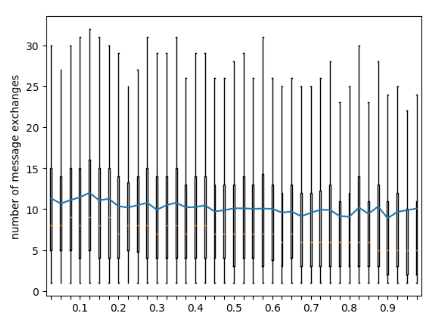
(b) M depending on initial distance, for $\varepsilon = 0.5$



(c) Messages per time unit with $\mathcal{A}_{lc}$, for $\varepsilon = 0.1$



(d) Messages per time unit with $\mathcal{A}_{lc}$, for $\varepsilon = 0.5$



(e) M depending on ε, for $d_0 = 400$

Fig. 21: Values in the 3D case when both players follow a continuous movement