Efficient Assignment of Identities in Anonymous Populations

Leszek Gąsieniec[1], Jesper Jansson[2], Christos Levcopoulos[3], and Andrzej Lingas[3]

[1] Department of Computer Science, University of Liverpool, Street, L69 38X, U.K. L.A.Gasieniec@liverpool.ac.uk

[2] Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan. jj@i.kyoto-u.ac.jp

[3] Department of Computer Science, Lund University, 22100 Lund, Sweden. {Christos.Levcopoulos, Andrzej.Lingas}@cs.lth.se

Abstract. We consider the fundamental problem of assigning distinct labels to agents in the probabilistic model of population protocols. In this distributed model, during each consecutive step the random scheduler draws uniformly at random a pair from the population of n identical agents. The two chosen agents interact and on the conclusion of the step they update their states according to the predefined transition function. This function is designed to allow agents to solve the considered shared computational task.

Our protocols operate under the assumption that the size n of the population is embedded in the transition function. In addition, our solutions rely on a unique leader which can be precomputed with a negligible impact on our upper bounds. The efficiency of our protocols is expressed in terms of the number of states utilized by agents, the size of the range from which the labels are drawn, and the expected number of interactions required by our solutions.

Among other things, we consider *silent* labeling protocols, where eventually each agent reaches its final state and remains in it forever, as well as *safe* labeling protocols which (i) can produce a valid agent labeling in a finite number of interactions, and (ii) guarantee that at any step of the protocol no two agents have the same label.

We first focus on labeling silent or safe protocols which use very small number of states and labels from range $1, \ldots, n$. We provide a silent and safe protocol which uses only $n + 5\sqrt{n} + 4$ states. The expected number of interactions required by the protocol is $O(n^3)$. On the other hand, we show that any safe protocol, as well as any silent protocol which provides a valid labeling with probability $> 1 - \frac{1}{n}$, uses at least $n+\sqrt{n}-1$ states. It follows that our protocol is almost state-optimal. In addition, we present a variant of this protocol which uses $n(1 + \varepsilon)$ states. The expected number of interactions required by this variation is $O(n^2/\varepsilon^2)$, where $\varepsilon = \Omega(n^{-1/2})$. On the other hand, we show that for any safe labeling protocol utilizing $n+t < 2n$ states the expected number of interactions required to achieve a valid labeling is at least $\frac{n^2}{t+1}$. We show also an analogous lower bound on the expected number of interactions for any silent labeling protocol which provides a valid labeling with probability 1.

Next, we present a fast labeling protocol for which the required number of interactions is asymptotically optimal, i.e., $O(n \log n)$, with high probability. It uses

$O(n)$ states and draws labels from the range $1, \ldots, 2n$. In addition, we provide a generalization of the protocol requiring $O(n \log n/\varepsilon)$ interactions with high probability, utilizing $(2+\varepsilon)n + O(\log n)$ states and drawing labels from the range $1, \ldots, (1+\varepsilon)n$, where $\varepsilon = \Omega(n^{-1})$. On the other hand, we consider a natural class of labeling, the so-called *pool protocols*, that includes our fast protocol and its generalization. We show that the expected number of interactions required by any pool protocol is at least $\frac{n^2}{r+1}$, when the label range is limited to $1, \ldots, n+r < 2n$. Our fast labeling protocols are also silent and safe.

1 Introduction

The problem of assigning and further maintaining unique identifiers for entities in distributed systems is one of the core problems related to network integrity. In addition, a solution to this problem is often an important preprocessing step for more complex distributed algorithms. The tighter the range that the identifiers are drawn from, the harder the assignment problem becomes.

In this paper we adopt the probabilistic population protocol model in which we study the problem of assigning to all agents distinct identifiers which we refer to as *labels*. The adopted model was originally intended to model large systems of agents with limited resources (state space) [5]. In this model the agents are prompted to interact with one another towards a solution of a shared task. The execution of a protocol in this model is a sequence of pairwise interactions between randomly chosen agents. During an interaction, each of the two agents: the *initiator* and the *responder* (the asymmetry assumed in [5]) update its state in response to the observed state of the other agent according to the predefined (global) transition function.

Designing our population protocols for the problem of assigning unique labels to the agents (labeling problem), we make a natural assumption that the number n of agents is known in advance. Our protocols would also work if only an upper bound on the number of agents is known to agents. In fact, in such case the problem becomes easier as the range from which the labels are drawn is larger. Our labeling protocols also use the concept of a *leader*, i.e., an agent singled out from the population, which improves coordination of more complex tasks and processes. A good example is synchronization via phase clocks propelled by leaders. More examples of leader-based computation can be found in [6].

In the original model of population protocol proposed and studied in [5], the state space of agents was assumed to be constant. Subsequently, by allowing the number of utilized states to grow with the number n of agents a larger number of problems became solvable in this model. Another important feature of this model is the space/time trade-off where larger state space can dramatically decrease the expected number of interactions needed to solve the considered task.

For instance, any protocol for the exact majority problem (where the bias between competing opinions is very small or equal to zero) utilizing a constant number of states results in the expected number of interactions $\Omega(n^2)$. In contrast, in the protocols utilizing $O(poly(\log n))$ states the expected number of interactions drops to $O(n \cdot poly(\log n))$ [20]. This is important as the expected number of interactions $O(n \log n)$ is a natural lower bound to solve any non-trivial problem by a population protocol. The main reason is that $\Omega(n \log n)$ interactions are needed to achieve a positive constant probability that each agent is involved in at least one interaction [11]. In fact, there is already an informally agreed class of fast population protocols which require $O(n \cdot poly(\log n))$ interactions. This class can be defined in terms of the notion of parallel time in probabilistic population protocols which refers to the total number of interactions divided by the number of agents n. Namely, fast population protocols have polylogarithmic parallel execution time. Similar development with respect to the space/time trade-off can be found in relation to leader election [1,10,12]. The newest results [14,22,23] elaborate on state-optimal leader election protocols utilizing $O(\log \log n)$ states. These include the fastest possible protocol [14] based on $O(n \log n)$ interactions in expectation, and a slightly slower protocol [22] requiring $O(n \log^2 n)$ interactions with high probability.

In the unique labeling problem adopted here, the number of utilized states needs to reflect the number of agents n. Perhaps the simplest protocol for unique labeling in population networks is as follows [17] (cf. [15]). Initially, all agents hold label 1 which is equivalent with all agents being in state 1. In due course, whenever two agents with the same label i interact, the responder updates own label to $i + 1$. The advantage of this simple protocol is that it does not need any knowledge of the population size n and it utilizes only n states and assigns labels from the smallest possible range $[1, n]$[4]. The severe disadvantage is that it needs at least a cubic in n number of interactions (getting rid of the last multiple label i, for all $i = 1, \ldots n - 1$, requires a quadratic number of interactions in expectation) to achieve the configuration in which the agents have distinct labels.

In the following two examples of protocols for unique labeling, we assume that the population size n is embedded in the transition function, such protocols are commonly used and known as non-uniform protocols [4], and one of the agents is distinguished as the leader, see leader based protocols [6].

In the first of the two examples, we instruct the leader to pass labels $n, n - 1, ..., 2$ to the encountered subsequently unlabeled yet agents and finally assign

[4] We shall denote a range $[p, \cdots, q]$ by $[p, q]$ further.

1 to itself. The protocol uses only $2n - 1$ states (n states utilized by the leader and $n - 1$ states by other agents) and it assigns unique labels in the smallest possible range $[1, n]$ to the n agents. Unfortunately, this simple protocol requires $\Omega(n^2 \log n)$ interactions because as more agents get their labels, interactions between the leader and agents without labels become less likely. The probability of such an encounter drops from $\frac{1}{n}$ at the beginning to $\frac{1}{n(n-1)}$ at the end of the process.

By using randomization, we can obtain a much faster simple protocol as follows. We let the leader to broadcast the number n to all agents. It requires $O(n \log n)$ interactions with high probability (w.h.p. for short) [20]. When an agent gets the number n, it uniformly at random picks a number in $[1, n^3]$ as its label. The probability that a given pair of agents gets the same label is only $\frac{1}{n^3}$. Hence, this protocol assigns unique labels to the agents with probability at least$1 - \frac{1}{n}$. It requires only $O(n \log n)$ interactions w.h.p. The drawback is that it uses $O(n^3)$ states and the large range $[1, n^3]$. This method also needs a large number of random bits independent for each agent.

Besides the efficiency and population size aspects, there are also other deep differences between the three examples of labeling protocols. An agent in the first protocol never knows whether or not it shares its label with other agents. This deficiency cannot happen in the case of the second protocol but it takes place in the third protocol although with a small probability,

In this paper we consider among other things *silent* labeling protocols and *safe* labeling protocols. We say that a (non-necessarily labeling) protocol is silent if eventually each agent reaches its final state and remains in it forever. We say that a labeling protocol is safe if, for any given set of agents: (i) there exists a finite run of the protocol that produces a valid agent labeling; and (ii) at any time step in any run of the protocol, no two agents have the same label. While the concept of a silent population protocol is well established in the literature [16], the concept of a safe labeling protocol is new. Note that a safe labeling protocol is partially correct in a strong sense since at any time step of its run, the assigned labels form a valid partial labeling. This in particular might be useful in the situation where the protocol has to be terminated before completion due to some unexpected emergency or running out of time.

Observe that among the three examples of labeling protocols, only the second one is both silent and safe. The first example protocol is silent [16] but not safe. Finally, the third (probabilistic) one is silent and almost safe as it violates the condition (ii) only with small probability.

Two natural questions arise under the assumption that the number n of agents is known at the beginning to exactly one of the agents only (an implicit leader).

1. Can one design a safe or silent protocol for the labeling problem utilizing substantially smaller number of states than $2n$ and possibly the minimal label range $[1, n]$?
2. Can one design a protocol for the labeling problem requiring an asymptotically optimal number of $O(n \log n)$ interactions w.h.p., utilizing $O(n)$ states and the label range of size $O(n)$?

We provide positive answers to both questions. We also discuss the relevant lower bounds.

We first provide a silent and safe protocol which uses only $n + 5\sqrt{n} + 4$ states and the label range $[1, n]$. The expected number of interactions required by the protocol is $O(n^3)$. On the other hand, we show that any safe labeling protocol, as well as any silent protocol which provides a valid labeling with probability larger than $1 - \frac{1}{n}$, uses at least $n + \sqrt{n} - 1$ states. It follows that our protocol is almost state-optimal. In addition, we present a variant of this protocol which uses $n(1+\varepsilon)$ states. The expected number of interactions required by this variation is $O(n^2/\varepsilon^2)$, where $\varepsilon = \Omega(n^{-1/2})$. On the other hand, we show that for any safe labeling protocol utilizing $n + t < 2n$ states the expected number of interactions required to achieve a valid labeling is at least $\frac{n^2}{t+1}$. We show also an analogous lower bound on the expected number of interactions for any silent protocol which provides a valid labeling with probability 1.

Next, we present a population protocol that w.h.p. requires an asymptotically optimal number of $O(n \log n)$ interactions to assign distinct labels from the range $[1, 2n]$. Only $O(n)$ states are used by the protocol. We also present a more involved generalization of the protocol, where the range of assigned labels is $[1, (1+\varepsilon)n]$. The generalized protocol requires $O(n \log n/\varepsilon)$ interactions in order to complete the assignment of distinct labels from $[1, (1+\varepsilon)n]$ to the n agents, w.h.p. It uses $(2+\varepsilon)n + O(\log n)$ states. Both protocols are silent and safe.

Finally, we consider a natural class of population protocols for the unique labeling problem, the so-called *pool protocols*, including our fast labeling protocols. We show that for any protocol in this class that picks the labels from the range $[1, n+r]$, the expected number of interactions is $\Omega(\frac{n^2}{r+1})$.

Importantly, a unique leader can be efficiently computed as a preprocessing to our protocols without substantially increasing our upper bounds, see the discussion in the third paragraph of subsection 1.2 (Related work).

Our results are summarized in Tables 1 and 2.

Theorem	# states	# interactions	Range
Theorem 1	$n + 5 \cdot \sqrt{n} + 4$	expected $O(n^3)$	$[1, n]$
Theorem 2	$(1 + \varepsilon)n$	expected $O(n^2/\varepsilon^2)$	$[1, n]$
Theorem 3	$O(n)$	$O(n \log n)$ w.h.p.	$[1, 2n]$
Theorem 4	$(2 + \varepsilon)n + O(\log n)$	$O(n \log n/\varepsilon)$ w.h.p.	$[1, (1 + \varepsilon)n]$

Table 1. Upper bounds on the number of states, the number of interactions and the range required by the safe labeling protocols presented in this paper. In Theorem 2, ε is $\Omega(n^{-0.5})$ while in Theorem 4.9 $\Omega(n^{-1})$.

Protocol type	# states	# interactions	Theorem
any*	n	$\Omega(n \log n)$ w.h.p.	Theorem 5
safe	$n + \sqrt{n} - 1$	-	Theorem 6 (1st part)
safe, $n + t < 2n$ states	-	expected $\frac{n^2}{t+1}$	Theorem 6 (2nd part)
silent**	$n + \sqrt{n} - 1$	-	Theorem 7 (1st part)
silent***, $n + t < 2n$ states	-	expected $\frac{n^2}{t+1}$	Theorem 7 (2nd part)
pool, range $[1, n + r]$	-	expected $\frac{n^2}{r+1}$	Theorem 8

Table 2. Lower bounds on the number of states or/and the number of interactions required by labeling protocols. (*) Any labeling protocol that is capable to produce a valid labeling. (**) The silent protocol in Theorem 7 (first part) is assumed to produce a valid labeling with probability greater than $1 - \frac{1}{n}$. (***) The silent protocol in Theorem 7 (2nd part) is assumed to produce a valid labeling with probability 1

.

1.1 The computational model of population protocols

There is given a population of n agents that can pairwise interact in order to change their states and in this way perform a computation. A population protocol can be formally specified by providing a set Q of possible states, a set O of possible outputs, a transition function $\delta : Q \times Q \to Q \times Q$, and an output function $o : Q \to O$. The current state $q \in Q$ of an agent is updated during interactions. Consequently, the current output $o(q)$ of the agent also becomes updated during interactions. The current state of the set of n agents is given by a vector in Q^n with the current states of the agents. A computation of a population protocol is specified by a sequence of pairwise interactions between agents. In every time step, an ordered pair of agents is selected for interaction by a probabilistic scheduler independently and uniformly at random. The first agent in the selected pair is called the initiator while the second one is called the responder.

The states of the two agents are updated during the interaction according the transition function δ.

We can specify a problem to solve by a population protocol by providing the set of input configurations, the set O of possible outputs, and the desired output configurations for given input configurations. For the unique labeling problem, all agents but one are initially in the same state q_0 while a single agent called the leader is in a special state corresponding to the number n. The set O is just the set of positive integers. A desired configuration is when all agents output their distinct labels. The *stabilization time* of an execution of a protocol is the number of interactions until the states of agents form a desired configuration from which no sequence of pairwise interactions can lead to a configuration outside the set of desired configurations.

1.2 Related Work

There are several papers concerning labeling of processing units (also known as renaming) in different communication models [18]. E.g., Berenbrink et al. [9] present efficient algorithms for the so-called lose and tight renaming in shared memory systems improving on or providing alternative algorithms to the earlier algorithms by Alistarh et al. [2,3]. The lose renaming where the label space is larger that the number of units is shown to admit substantially faster algorithms than the tight renaming [3,9].

The problem of assigning unique labels to agents has been studied in the model of population protocols solely in the works of Beauquier et al. [8,15]. In [15], the emphasis is on estimating the minimum number of states which are required by apparently non-safe protocols. In [8], the authors provide among other things a generalization of a leader election protocol to include a distribution of m labels among n agents, where $m \leq n$. In the special case of $m = n$, all agents will receive unique labels. No analysis on the number of interactions required by the protocol is provided in [8]. Their focus was on the feasibility of the solution, i.e., that the considered process eventually stabilizes in the final configuration. Their protocol seems inefficient in the state space aspect as it needs many states/bits to keep track of all the labels.

The labeling problem has been also studied in the context of self-stabilizing protocols where the agents start in arbitrary (not predefined) states, see [16,17]. In [17], Cai et al. propose a solution which coincides with our first example of labeling protocols presented in the introduction. In a very recent work [16], Burman et al. study both slow and fast labeling protocols focusing mainly on the asymptotic bounds, and with the latter utilizing exponential number of states. The protocols in both papers require the exact knowledge of n and they are not safe. In our paper, thanks to the predefined leader in addition to the safety and silence properties we also obtain tight exact results on the number of utilized states and labels. Please note that by utilizing extra $O(\log\log n)$ initialization states and leader election results from [14] and [22] one can compute and confirm the leader in $O(n\log n)$ iterations with constant probability and in $O(n\log^2 n)$ iterations with high probability. However, if a protocol can utilize $\Theta(n^c)$ states, for any constant $c < 1$, a unique leader can be elected with high probability in $O(n\log n)$ iterations, as described in [11,19]. In other words, one can precede our labeling protocols with leader election which makes a negligible impact on the main results in this paper.

The most closely related problem more studied in the literature is that of counting the population size, i.e., the number of agents. It has been recently studied by Aspnes et al. in [7] and Berenbrink et al. in [11]. We assume that the population size is initially known to one of the agents. Alternatively, it can be computed by using the protocol counting the exact population size given in [11]. The aforementioned protocol computes the population size in $O(n\log n)$ interactions w.h.p., using $\tilde{O}(n)$ states. Another possibility is to use the protocol computing the approximate population size, presented in [11]. The latter protocol requires $O(n\log^2 n)$ interactions to compute the approximate size w.h.p., and it uses only a poly-logarithmic number of states. For references to earlier papers on protocols for counting or estimating the population size, see [11]. To combine a protocol for counting population size with our protocols for unique labeling assuming the knowledge of n, one needs to run a protocol for leader typically being already a part of the former protocol. There is a vast literature on population protocols for leader election [14,19,20,22]. For the aforementioned purpose, the most relevant is the recent time- and state-optimal leader election protocol due to Berenbrink et al. requiring $O(n\log n)$ interactions and $O(\log\log n)$ states [14] (see also [22]). Our population protocols for unique labeling use the known population protocol for (one-way) epidemics, or broadcasting. It completes spreading a message in $\Theta(n\log n)$ interactions w.h.p. and it uses only two states [20] (see also Fact 4).

1.3 Organization of the Paper

In the next section, we provide basic facts on probabilistic inequalities and population protocols for broadcasting and counting. Section 3 is devoted to the almost state-optimal safe protocol with the label range $[1, n]$ and its variation. In Section 4, we present our fast safe protocol for unique labeling in the range $[1, 2n]$ and its generalization to include the range $[1, n(1+\varepsilon)]$. Section 5 is devoted to lower bounds on the number of states or the number of interactions for safe, silent and the so-called pool protocols for unique labeling. We conclude with final remarks

2 Preliminaries

2.1 Probabilistic bounds

Fact 1 *(The union bound) For a sequence* $A_1,\ A_2, ..., A_r$ *of events,* $Prob(A_1 \cup A_2 \cup ......A_r) \leq \sum_{i=1}^{r} Prob(A_i)$.

Fact 2 *(multiplicative Chernoff lower bound) Suppose* $X_1, ..., X_n$ *are independent random variables taking values in* $\{0, 1\}$. *Let X denote their sum and let* $\mu = E[X]$ *denote the sum's expected value. Then, for any* $\delta \in [0, 1]$,
$Prob(X \leq (1 - \delta)\mu) \leq e^{\frac{\delta^2 \mu}{2}}$ *holds. Similarly, for any* $\delta \geq 0$, $Prob(X \leq (1 + \delta)\mu) \leq e^{\frac{\delta^2 \mu}{2+\delta}}$ *holds.*

Fact 3 *[20] For all* $C > 0$ *and* $0 < \delta < 1$, *during* $Cn \log n$ *interactions, with probability at least* $1 - n^{-O(\delta^2 C)}$ *, each agent participates in at least* $2C(1 - \delta) \log n$ *and at most* $2C(1 + \delta) \log n$ *interactions.*

2.2 Broadcasting and counting

We shall refer to the following broadcast process which can be completed during $\Theta(n \log n)$ interactions w.h.p. Each agent is either in a sate of M-type (got the message) or in a state of ¬M-type. Whenever an agent in a state of M-type interacts with an agent in a state of ¬M-type, the latter changes its state to a state of M-type (gets the message). The process starts when the first agent gets the message and completes when all agents have the message.

Fact 4 *There is a constant* c_0 *, such that for* $c \geq c_0$, *the broadcast process completes in* $cn \log n$ *interactions with probability at least* $1 - n^{-\Theta(c)}$.

Berenbrink et al. [11] obtained among other things the following results on counting the population size, i.e., the number of agents.

Fact 5 *There is a protocol for a population of an unknown number* n *of agents such that w.h.p., after* $O(n \log^2 n)$ *interactions the protocol stabilizes and each agent holds the same estimation of the population size which is either* $\lceil \log n \rceil$ *or* $\lfloor \log n \rfloor$. *The protocol uses* $O(\log^2 n \log \log n)$ *states.*

Fact 6 *There is a protocol for a population of an unknown number* n *of agents such that w.h.p., after* $O(n \log n)$ *interactions the protocol stabilizes and each agent holds the exact population size. The protocol uses* $\tilde{O}(n)$ *states.*

3 State- and range-optimal labeling

In this section we propose and analyze state-optimal safe protocols which utilize labels from the smallest possible range $[1, n]$. We assume that the number of agents n is known and the leader is given. We propose a safe labeling protocol *Single-Cycle* which utilizes $n + 5\sqrt{n} + 4$ states and the expected number of interactions required by the protocol is $O(n^3)$. We show later that any safe protocol operating under the adopted assumptions requires $n + \sqrt{n} - 1$ states, see Theorem 6, indicating that this protocol is almost state-optimal. Finally we propose a partial parallelization of *Single-Cycle* protocol called k-*Cycle* protocol which utilizes $(1 + \varepsilon)n$ states and $O((n/\varepsilon)^2)$ interactions for $\varepsilon = \Omega(n^{-1/2})$.

3.1 Labeling protocol

The main idea behind the state efficient labeling protocol is to use two agents: the initial leader A and a nominated (by A) agent B, as partial *label dispensers*. These two agents jointly dispense unique labels for the remaining *free* (non-labeled yet) agents in the population where agent A dispenses the first and agent B the second part of each individual label. For the simplicity of presentation, we assume that n is a square of some integer. During execution of the protocol agent A uses partial labels $\texttt{label}(a) \in \{0, \ldots, \sqrt{n} - 1\}$ and B uses partial labels $\texttt{label}(b) \in \{1, \ldots, \sqrt{n}\}$. The two dispensers label every agent by a unique pair of partial labels $(\texttt{label}(a), \texttt{label}(b))$ where the combination (i, j) is interpreted as the integer label $i \cdot \sqrt{n} + j$. The protocols labels first all *free* (different to dispensers unlabeled) agents and eventually give labels $(0, 2)$ to agent B and $(0, 1)$ to agent A.

In a nutshell, the labeling process is based on a single cycle of interactions between dispensers A and B and the free agents. Agent A awaits an interaction with a free agent F when A dispenses to F its current partial label $\texttt{label}(a)$. Now F awaits an interaction with B in order to receive the second part of its label. And when this happens agent F concludes with the combined label and agent B awaits an interaction with A to inform that the next free agent needs to be labeled. On the conclusion of this interaction if $\texttt{label}(b) > 1$ agent B adopts new partial label $\texttt{label}(b) - 1$, otherwise B adopts $\texttt{label}(b) = \sqrt{n}$ and agent A adopts new label $\texttt{label}(a) - 1$. The only exception is when $\texttt{label}(a) = 0$ and $\texttt{label}(b) = 2$ when agent B adopts label $(0, 2)$ and agent A adopts label $(0, 1)$ and both agents conclude the labeling process. **State utilization in *Single-Cycle* protocol**

[Agent A] Since $\texttt{label}(a) \in \{0, \ldots \sqrt{n} - 1\}$ dispenser A utilizes $2 \cdot \sqrt{n} + 2$ states including:

- $A.\texttt{init} = (1)$ **# the initial (leadership) state of dispenser A,**
- $A[\texttt{label}(a), \texttt{await}(F)]$ **# dispenser A carrying partial label $\texttt{label}(a)$ awaits interaction with a free agent F,**
- $A[\texttt{label}(a), \texttt{await}(B)]$ **# dispenser A carrying partial label $\texttt{label}(a)$ awaits interaction with dispenser B,**
- $A.\texttt{final} = (0, 1)$ **# the final state of A.**

[Agent B] Since $\texttt{label}(b) \in \{0, \ldots \sqrt{n}\}$ dispenser B utilizes $2 \cdot \sqrt{n} + 3$ states including:

- $B[\texttt{label}(b), \texttt{await}(F)]$ **# dispenser B carrying partial label $\texttt{label}(b)$ awaits interaction with a free agent F,**
- $B[\texttt{label}(b), \texttt{await}(A)]$ **# dispenser B carrying partial label $\texttt{label}(b)$ awaits interaction with dispenser A**
- $B.\texttt{final} = (0, 2)$ **# the final state of B.**

[Agent F] Since free agents carry partial labels $\texttt{label}(a) \in \{0, \ldots \sqrt{n}-1\}$ and eventually adopt one of the $n - 2$ destination labels (excluding dispensers) they utilize $n + \sqrt{n} - 1$ states including:

- $F.\texttt{init} = (0)$ **# the initial (non-leader) sate of F**
- $F[\texttt{label}(a), \texttt{await}(B)]$ **# free agent F carrying partial label $\texttt{label}(a)$ awaits interaction with dispenser B,**
- $F.\texttt{final} = (\texttt{label}(a), \texttt{label}(b))$ **# the final state of F.**

In total *Single-Cycle* protocol requires $n + 5 \cdot \sqrt{n} + 4$ states.

Transition function in *Single-Cycle* protocol

[Step 0] Initialization During the first interaction of A with a free agent the second dispenser B is nominated. Both dispensers adopt their largest labels. Agent A awaits a free agent in the initial state while agent B awaits a free agent carrying a partial label obtained from A.

- $(A.\texttt{init}, F.\texttt{init})$
  $\rightarrow (A[\texttt{label}(a) = \sqrt{n}-1, \texttt{await}(F)], B[\texttt{label}(b) = \sqrt{n}, \texttt{await}(F)]),$

The three steps C_1, C_2, and C_3 of the labeling cycle are given below.
[Step C_1] Agent A dispenses partial label During an interaction of agent A with a free agent F the current partial label $\texttt{label}(a)$ is dispensed to F. Both agents await interactions with dispenser B which is ready to interact with partially labeled F but not A.

- $(A[\texttt{label}(a), \texttt{await}(F)], F.\texttt{init})$
  $\rightarrow (A[\texttt{label}(a), \texttt{await}(B)], F[\texttt{label}(a), \texttt{await}(B)])$ **# Go to Step C_2**

[Step C_2] Agent B dispenses partial label During an interaction of agent B with a free agent F which carries partial label $\texttt{label}(a)$, the complementary current partial label $\texttt{label}(b)$ is dispensed to F. Agent F concludes in the final state with the combined label $(\texttt{label}(a), \texttt{label}(b))$. Agent B is now ready for interaction with A.

- $(B[\texttt{label}(b), \texttt{await}(F)], F[\texttt{label}(a), \texttt{await}(B)])$
  $\rightarrow (B[\texttt{label}(b), \texttt{await}(A)], F.\texttt{final} = (\texttt{label}(a), \texttt{label}(b)))$ **# Go to Step C_3**

[Step C_3] Agent A and B negotiate a new label or conclude In the case when $\texttt{label}(a) = 0$ and $\texttt{label}(b) = 2$ the dispensers A and B conclude in states $(0, 1)$ and $(0, 2)$ respectively, see the first transition. Otherwise a new combination of partial labels is agreed and the protocol goes back to Step C_1.

- $(A[\texttt{label}(a) = 0, \texttt{await}(B)], B[\texttt{label}(b) = 2, \texttt{await}(A)])$
  $\rightarrow (A.\texttt{final} = (0, 1), B.\texttt{final} = (0, 2))$ **# Conclude the labeling process**
- $(A[\texttt{label}(a) = 0, \texttt{await}(B)], B[\texttt{label}(b) > 2, \texttt{await}(A)])$ **or**
  $(A[\texttt{label}(a) > 0, \texttt{await}(B)], B[\texttt{label}(b) > 1, \texttt{await}(A)])$
  $\rightarrow (A[\texttt{label}(a), \texttt{await}(F)], B[\texttt{label}(b) - 1, \texttt{await}(F)])$ **# Go to Step C_1**
- $(A[\texttt{label}(a) > 0, \texttt{await}(B)], B[\texttt{label}(b) = 1, \texttt{await}(A)])$
  $\rightarrow (A[\texttt{label}(a) - 1, \texttt{await}(F)], B[\texttt{label}(b) = \sqrt{n}, \texttt{await}(F)])$ **# Go to Step C_1**

Theorem 1. Single-cycle *protocol is silent and safe, it utilizes* $n + 5 \cdot \sqrt{n} + 4$ *states and the minimal label range* $[1, n]$. *The expected number of interactions required by the protocol is* $O(n^3)$.

Proof. The protocol is silent by its definition. It is also safe as all labels are dispensed in the sequential manner and the labeling process concludes when the two dispensers finalize their own labels. In particular, as soon as the two dispensers A and B are established they operate in a short cycle formed of steps $C1, C2$ and $C3$ labeling one by one all free agents in the population. One can observe that the sequence of cycles mimics the structure of two nested loops where the external loop iterates along the partial labels of A and the internal one along partial labels of B. In total, we have $n - 2$ iterations where the expected number of interactions required by each iteration is $O(n^2)$. Thus one can conclude that the expected number of interactions required by the whole labeling process is $O(n^3)$. By the definition of the protocol the range of assigned labels is $[1, n]$. Finally, as indicated earlier in this section the number of all states utilized by the protocol is equal to $n + 5 \cdot \sqrt{n} + 4$. □

Observe that when the exact value of n is embedded in the transition function on the conclusion all agents become dormant, i.e., they stop participating in the labeling process. One could redesign the protocol such that the labels are dispensed in the increasing order using a diagonal method where agent A gets label $(0,0)$, agent B gets label $(0,1)$, the first labeled free agent gets $(1,0)$, the second $(0,2)$, then $(1,1)$ and $(2,0)$, when A and B start using the next diagonal, etc. In this case the size of the population does not need to be known in advance, however, the two dispensers will never stop searching for free agents yet to be labeled.

Faster Labeling We observe that one can partially parallelize *Single-Cycle* protocol by instructing leader A to form k pairs of dispensers where each pair labels agents in a distinct range of size n/k. In such case the new k*-cycle* protocol requires extra $2k$ states to allow leader A initialize the labeling process (create two dispensers) in all k cycles. Thus the total number of states is bounded by $n+2k+k\cdot(5\sqrt{n/k}+4) = n+6k+5k\cdot\sqrt{n/k} < n+6(k+\sqrt{nk}) < n+12\sqrt{nk}$, as $k < \sqrt{nk}$. As we need to pick k for which $n+12\sqrt{nk} \leq n+n\varepsilon$ we conclude that $k \leq n\varepsilon^2/144$.

One can show that for $k = n\varepsilon^2/144$, the expected number of interactions required by the k*-cycle* protocol is $O(n^2/\varepsilon^2)$. Note that in order to initialize k cycles the leader A has to communicate with $2k-1$ free agents. As k is at most a small fraction of n during the search for dispensers for each cycle the number of free agents is always greater than $n/2$ (in fact it is very close to n). Thus the probability of forming a new dispenser during any interaction is greater than $1/2n$, i.e., the product of the probability $1/n$ that the random scheduler selects leader A as the initiator, times the probability greater than $1/2$ that the responder is a free agent. In order to finish the initialization, we need to create new dispensers $2k-1$ times. Using Chernoff bound, we observe that after $O(kn) = O(n^2/\varepsilon^2)$ interactions all k cycles have their two dispensers formed. As each cycle dispenses $n/k = 144/\varepsilon^2$ labels and the expected number of interactions required to dispense a single label is $O(n^2)$ with high probability, the expected number of interactions required by a specific cycle to generate all labels is $O(n^2/\varepsilon^2)$ also with high probability. Hence, the expected number of interactions required to conclude the labeling process is $O(n^2/\varepsilon^2)$. Finally, note that for small values of ε approaching $n^{-1/2}$ k*-cycle* protocol reduces to *Single-cycle* protocol.

Theorem 2. *For* $k = n\varepsilon^2/144$*, where* $\varepsilon = \Omega(n^{-1/2})$*, and the minimal label range* $[1,n]$*, the proposed* k-cycle *labeling protocol provides a space-time trade-off in which utilization of* $(1+\varepsilon)n$ *states permits the expected number of interactions* $O(n^2/\varepsilon^2)$.

4 Labeling with asymptotically optimal number of interactions, nearly optimal number of states and range

In this section, we provide a silent and a safe labeling protocol that assigns unique labels from the range $[1, 2n]$ to n agents in $O(n \log n)$ interactions w.h.p. Then, we generalize the protocol to include the range $[1, (1+\varepsilon)n]$. We show that the generalized protocol assigns unique labels from $[1, (1+\varepsilon)n]$ in $O(n \log n/\varepsilon)$ interactions w.h.p. In the first protocol, the agents use $O(n)$ states, in the second protocol only $(2 + \varepsilon)n + O(\log n)$ states.

4.1 Range $[1, 2n]$

The protocol runs in two main phases. The idea of the first phase resembles that of load balancing [11], the difference is that tokens (in our case labels and interval sub-ranges) are distinct.

We assume that at the beginning of the first phase, a leader agent knows the number n of agents in the population network. The leader assigns the label 1 and also temporarily the interval $[2, n]$ to itself. Next, whenever two agents interact, one with label and a temporarily assigned interval $[q, r]$ where $r > q$ and the other without label, the former agent shrinks its interval to $[q, \lfloor \frac{q+r}{2} \rfloor]$ and it gives away the label $\lfloor \frac{q+r}{2} \rfloor + 1$ and if $\lfloor \frac{q+r}{2} \rfloor + 2 \leq r$ also the sub-interval $[\lfloor \frac{q+r}{2} \rfloor + 2, r]$ to the latter agent. Furthermore, whenever an agent with label and a temporarily assigned singleton interval $[q, q]$ interacts with an agent without label, the former agent cancels its interval and gives the label q to the latter agent. In the remaining cases, interactions have no effect. Note that during the first phase a sub-tree of the binary tree of the partition of the start interval $[1, n]$ with n leaves determined by the protocol rules is formed, see Fig. 1. Also observe that when an agent at an intermediate node of the tree interacts with an agent without label then the former agent migrates to the left child of the node while the latter agent lands at the right child of the node.

In the second phase, when an agent with a label $i \in [1, n]$ at a leaf of the tree interacts with an agent without label for the first time then the latter agent gets the label $i + n$. Interactions between agents (if any) at intermediate nodes of the tree and agents without labels are defined as in the first phase. The following lemma is central in showing that $O(n \log n)$ interactions are sufficient w.h.p. to implement our protocol.

Lemma 1. *There is a constant c such that after $cn \log n$ interactions in the first phase the number of agents without labels drops below $n/4$ w.h.p.*

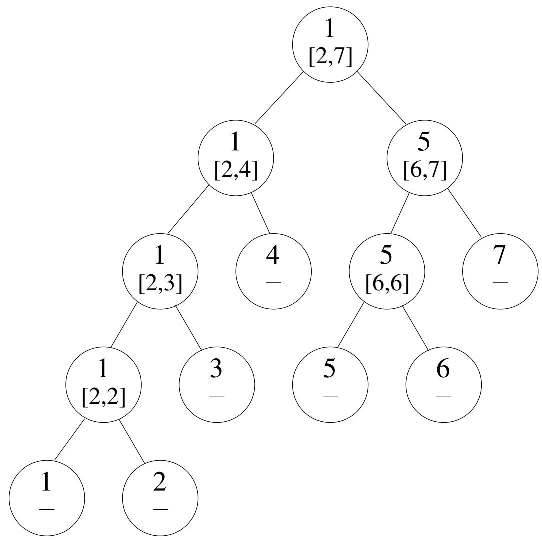

Fig. 1. An example of the partition tree of the start interval.

Proof. The proof is by contradiction. Suppose that a set F of at least $n/4$ agents without labels survives at least $cn \log n$ interactions, where the constant c will be specified later.

Consider first the leader agent starting with the interval $[2, n]$ during the aforementioned interactions. When the agent interacts with an agent without label its interval is roughly halved. We shall call such an interaction a success. The probability of success is at least $\frac{1}{4n}$. The expected number of successes is at least $\frac{c}{4} \log n$. By using Chernoff multiplicative bound given in Fact 2, we can set c to enough large constant so the probability of at least $\log_2 n + 1$ successes will be at least $1 - \frac{1}{n^2}$. This means that the leader will end up without any interval with so high probability during the $cn \log n$ interactions. The leader chooses the leftmost path in the binary partition tree of the start interval $[1, n]$. Consider an arbitrary path P from the root to a leaf in the tree. Note that several agents during distinct interactions can appear on the path. Define as a success an interaction in which an agent currently on P interacts with an agent without label. The expected number of successes is again at least $\frac{c}{4} \log n$ and again we can conclude that there are at least $\log_2 n + 1$ successes with probability at least $1 - \frac{1}{n^2}$. Simply, the probabilities of interacting with an agent without label are the same for all agents with labels, i.e., on some paths in the tree. Another way to argue is that the leader could make other decisions as to which roughly half of interval to preserve and the path choice. By the union bound (Fact 1), we conclude that all the n paths from the root to the leaves in the tree could be developed during the $cn \log n$ interactions, so all agents would get a label, with probability at least $1 - \frac{1}{n}$. We obtain a contradiction with the so long existence of the set F. $\square$

Lemma 2. *If the second phase starts after $cn \log n$ interactions, where c is the constant from Lemma 1, then only $O(n \log n)$ interactions are needed to assign labels in $[1, 2n]$ to the remaining agents without labels, w.h.p.*

Proof. The number of agents without labels at the beginning of the second phase is at most $n/4$ w.h.p. Hence, at the beginning of this phase the number of agents with labels is at least $\frac{3}{4}n$ w.h.p. An agent with label $i \leq n$ at a leaf of the tree can give the label $i + n$ to an agent without label only once. Since this can happen at most $\frac{n}{4}$ times, the number of agents with labels in $[1, n]$ that can give a label is always at least $\frac{n}{2}$ w.h.p. We conclude that for an agent without label the probability of an interaction with an agent that can give a label is is at least almost $\frac{1}{2n}$. Hence, after each $O(n)$ interactions the expected number of agents without label halves. It follows that the expected number of such interactions rounds is $O(\log n)$. Consequently, the number of the rounds is also $O(\log n)$ w.h.p. by Chernoff bound (Fact 2).

An alternative way to obtain the $O(n \log n)$ bound on the number of interactions w.h.p. is to use Fact 3 with $C = O(\frac{1}{1/2})$ and $\delta = \frac{1}{2}$. Then, each agent will interact with at least $C \log n$ agents w.h.p. during $Cn \log n$ interactions. Consequently, the probability that a given agent does not interact with any agent that can give a label during the aforementioned interactions is $(1 - \frac{1}{2})^{O(2) \log n}$. Hence, by picking enough large C, we conclude that each agent (in particular without label) will interact with at least one agent that can give a label during the $Cn \log n$ interactions w.h.p. $\square$

Lemma 3. *During both phases, no pair of agents gets the same label.*

Proof. The uniqueness of the label assignments in the first phase follows from the disjointedness of the labels and intervals assigned to agents before and after each interaction. This argument also works for the labels not exceeding n assigned later in the second phase. Finally, the uniqueness of the labels of the form $i + n$ follows from the uniqueness of the labels of the agents passing these labels. $\square$

Theorem 3. *There is a silent and safe protocol for population of n agents that w.h.p. assigns unique labels in the range $[1, 2n]$ to the agents equipped with $O(n)$ states in $O(n \log n)$ interactions.*

Proof. The correctness of label assignment in both phases and the fulfilling the condition (ii) in the definition of a safe protocol follows from Lemma 3. Both phases require $O(n \log n)$ interactions w.h.p. by Lemmata 1, 2.

To put the two phases described in Lemmata 1, 2 together, we let the leader agent to count its interactions. When the number of interactions of the leader in the first phase exceeds an appropriate multiplicity of $\log n$, the total number of interactions in the first phase achieves the required lower bound from Lemma 1 w.h.p. by Fact 3. Therefore, then the leader starts broadcasting the message on the transition to the second phase to the other agents. By Fact 4, the broadcasting increases the number of interactions only by $O(n \log n)$ w.h.p. (The leader can also stop the second phase in a similar fashion.) It follows in particular that the condition (i) in the definition of a safe protocol is satisfied.

To save on the number of states, instead of having states corresponding to all possible sub-intervals of $[1, n]$, we consider states corresponding to the nodes of the interval partition tree (see Fig. 1) whose sub-tree is formed in the first phase. More precisely, we associate two states with each intermediate node of the binary tree on n leaves and $n-1$ intermediate nodes. They indicate whether or not the agent at the intermediate node has already received the message about the transition to the second phase. Next, we associate four states to each leaf of the tree. They indicate similarly whether or not the agent at the leaf has already received the phase transition message and whether or not the agent has already passed a label to an agent without label in the second phase, respectively. With each label in the range $[n+1, 2n]$, we associate only a single state. Additionally, there are $O(\log n)$ states used by the leader to count interactions in order to start the second phase. Thus the total number of states does not exceed $2n + 4n + n + O(\log n)$. □

By combining the protocol of Theorem 3 with that of Berenbrink et al. for exact counting the population size (Fact 6), we obtain the following corollary on unique labeling when the population size is unknown to agents initially.

Corollary 1. *There is a silent and a safe protocol for a population of n agents that assigns unique labels in the range $[1, 2n]$ to the agents initially not knowing the number n, equipped with $\tilde{O}(n)$ states, in $O(n \log n)$ interactions w.h.p.*

Proof. We run first the protocol for exact counting (Fact 6) and then our protocol for unique labeling (Theorem 3) using the leader elected by the counting protocol. We can synchronize the three protocols in a similar fashion as we synchronized the two phases of our protocol additionally using $O(n \log n)$ interactions and $O(\log n)$ states. □

By using the method of approximate counting from [11] (Fact 5) instead of that for exact counting (Fact 6), we can decrease the number of states to $O(n)$

at the cost of increasing the label range to $[1, 8n]$ and the number of interactions required to $O(n \log^2 n)$.

4.2 Range $[1, (1+\varepsilon)n]$

The new protocol is obtained by the following modifications in the previous one. The leader which counts the number of own interactions starts broadcasting the phase transition message when the number of agents without labels drops below $n\varepsilon/4$ w.h.p. (see Lemma 4). The information about the transition to the second phase affects only the agents at the leaves of the interval partition tree, corresponding to labels in $[1, n\varepsilon]$. When they get the message about the phase transition, they know that they can pass a label which is the sum of their own label and n to the first agent without label they interact with. For this reason, only the agents at the leaves corresponding to labels in $[1, n\varepsilon]$ as well as the agents that are at the nodes that are ancestors of the aforementioned leaves participate in the broadcasting of the phase transition message. (Observe that the number of agents at these ancestors is $O(n\varepsilon)$ and an agent at such an ancestor also has a label in $[1, n\varepsilon]$.) In the second phase, besides the agents at the leaves corresponding to labels in $[1, n\varepsilon]$ and the agents without labels, also the agents at the intermediate nodes of the tree (if any) can really interact, in fact as in the first phase.

The following generalization of Lemma 1 is straightforward.

Lemma 4. *Let c be the constant from the statement of Lemma 1. During* $cn \log n/\varepsilon$ *interactions in the first phase the number of agents without label drops below* $n\varepsilon/4$ *w.h.p.*

Proof. The proof is a generalization of that for Lemma 1. Define F_ε as a set of at least $\varepsilon n/4$ agents without labels that survive at least $cn \log /\varepsilon$ interactions in the first phase. Note that for an arbitrary agent, the probability of interaction with a member in F_ε is at least $\frac{\varepsilon}{4n}$. The rest of the proof is analogous to that of Lemma 1. It is sufficient to replace F by F_ε and the probability $\frac{1}{4n}$ of an interaction with a member in F with that $\frac{\varepsilon}{4n}$ of an interaction with a member in F_ε. □

Having Lemma 4, we can easily generalize Lemma 2 to the following.

Lemma 5. *If the second phase starts after* $cn \log n/\varepsilon$ *interactions, where* c *is the constant from Lemmata 1, 4 , then only* $O(n \log n/\varepsilon)$ *interactions are needed to assign labels in* $[1, (1+\varepsilon)n]$ *to the remaining agents without labels, w.h.p.*

Proof. The number of agents without labels at the beginning of the second phase is smaller than $\varepsilon n/4$ w.h.p. Hence, at the beginning of the second phase the number of agents with labels in the range $[1, \varepsilon n]$ is at least $\frac{3\varepsilon n}{4}$ w.h.p. Recall that such an agent at a leaf of the tree can give a label to an agent without label only once. It follows that the number of agents with labels in $[1, \varepsilon n]$ that can give a label to an agent without label is always at least $\frac{\varepsilon n}{2}$ w.h.p. We conclude that for an agent without label the probability of an interaction with an agent that can give a label is at least almost $\frac{\varepsilon}{2n}$. Hence, after each $O(n/\varepsilon)$ interactions the expected number of agents without labels halves. It follows that the expected number of such interactions rounds is $O(\log n)$. Consequently, the number of the rounds is also $O(\log n)$ w.h.p. by Fact 2.

An alternative way to obtain the $O(n \log n/\varepsilon)$ bound on the number of interactions w.h.p. is to use Fact 3 analogously as in the proof of Lemma 2. The difference is that C is set to $O(\frac{2}{\varepsilon})$ instead of $O(2)$ since the set of agents that can give a label is of size at least $\frac{n\varepsilon}{2}$ now. □

We also need the following auxiliary lemma on broadcasting constrained to a subset of agents.

Lemma 6. *The leader can inform $\Theta(n\varepsilon)$ agents with labels not exceeding $O(n\varepsilon)$ about the phase transition using only these agents in $O(n \log n/\varepsilon)$ interactions.*

Proof. During the initial part of the broadcasting process, after every $O(n/\varepsilon)$ interactions, the expected number of agents participating in the broadcasting process doubles. Hence, after $O(n \log /\varepsilon)$ interactions, the expected number of informed agents will be $\Omega(n\varepsilon)$. Then, the expected number of uninformed agents will be halved for every $O(n/\varepsilon)$ interactions. So the expected number of rounds, each consisting of $O(n/\varepsilon)$ interactions, needed to complete the broadcasting is $O(\log n)$. It remains to turn the latter bound to a w.h.p. one. This can be done by using the Chernoff bounds (Fact 2).

Alternatively, we can define for the purpose of the analysis of the doubling part, a binary broadcast tree. An informed agent at an intermediate node of the tree after an interaction with an uninformed agent moves to a child of the node while the other agent now informed places at the other child (cf. the partition tree in the proofs of Lemmata 1, 4). Then, we can use the technique from the proofs of Lemmata 1, 4 to show that only $O(n \log n/\varepsilon)$interactions are required w.h.p. to achieve a configuration where only a constant fraction of the agents participating in the broadcasting is uninformed. To derive the same asymptotic upper bound on the number of interactions required by the halving part w.h.p., we can use Fact 3 with $C = O(\varepsilon^{-1})$ analogously as in the proofs of Lemmata 2, 5. □

The proof of the following theorem is analogous to that of Theorem 3 with Lemmata 1, 2 replaced by Lemmata 4, 5.

Theorem 4. *Let $\varepsilon > 0$. There is a silent and safe protocol for a population of n agents that assigns unique labels in the range $[1, (1+\varepsilon)n]$ to n agents equipped with $(2+\varepsilon)n + O(\log n)$ states in $O(n \log n/\varepsilon)$ interactions w.h.p.*

Proof. The distinctness of the labels assigned in both phases and fulfillment of the condition (ii) in the definition of a safe protocol follows by the same arguments as in the proof of Lemma 3.

By Lemmata 4, 5, both phases require $O(n \log n/\varepsilon)$ interactions w.h.p. The broadcasting about the phase transition starts when the number of agents without labels in the first phase drops below $n\varepsilon/4$ w.h.p. By Lemma 6, it requires $O(n \log n/\varepsilon)$ interactions w.h.p. since only the $\Theta(n\varepsilon)$ agents in states corresponding to labels in $[1, n\varepsilon]$ are involved in it. It follows in particular that the condition (i) in the definition of a safe protocol is satisfied.

The estimation of the number of needed states is more subtle than in Theorem 3. With each intermediate node of the interval partition tree that does not correspond to a label in $[1, n\varepsilon]$ (equivalently, that is not an ancestor of a leaf corresponding to a label in $[1, n\varepsilon]$), we associate a single state. (Recall here that if an agent at an intermediate node of the tree encounters an agent without label then the former agent moves to the left child of the node.) With each intermediate node corresponding to a label in $[1, n\varepsilon]$, we associate two states. They indicate whether or not the agent at the node has already got the message about phase transition. Next, with each leaf of the tree corresponding to a label i in $[1, n\varepsilon]$, we associate four states. They indicate whether or not the agent at the leaf has already got the message about the phase transition, and whether or not the agent has already passed the label $i + n$ to some agent without label, respectively. To each of the remaining leaves, we associate only a single state.

We also need $O(\log n/\varepsilon)$ additional states for the leader to count the number of own interactions in order to start broadcasting the message on transition to phase two at a right time step. In fact, we can get rid of the $O(\frac{1}{\varepsilon})$ factor here by letting the leader to count approximately each $\Theta(1/e)$ interaction. Simply, the leader can count only interactions with agents which have got labels not exceeding $O(\varepsilon n)$.

Finally, we have $n\varepsilon$ states corresponding to the labels in $[n+1, (1+\varepsilon)n]$. Thus, totally only $(2 + O(\varepsilon))n + O(\log n)$ states are sufficient. To get rid of the constant factor at ε, it is sufficient to run the protocol for a smaller $\varepsilon' = \Omega(\varepsilon)$. It does not change the asymptotic upper bound on the number of required interactions w.h.p. and even it decreases the range of the labels. □

Note that ε in Theorem 4 does not have to be a constant, it can be even so small as $O(n^{-1})$.

By combining the protocol of Theorem 4 with that of Berenbrink et al. for exact counting the population size (Fact 6), we obtain the following corollary on unique labeling when the population size is unknown to agents initially. The proof is analogous to that of Corollary 1.

Corollary 2. *Let $\varepsilon > 0$. There is a silent and safe protocol for a population of n agents that assigns unique labels in the range $[1, (1+\varepsilon)n]$ to the agents initially not knowing the number n, equipped with $\tilde{O}(n)$ states in $O(n \log n/\varepsilon)$ interactions w.h.p.*

5 Lower bounds

In this chapter, we derive several lower bounds in the number of states or the number of interactions required by safe, silent or the so-called pool protocols for unique labeling. Importantly, these lower bounds also hold in our model assuming that the population size is known to exactly one of the agents initially.

The following general lower bound valid for any range of labels follows immediately from the definitions of a population protocol and the problem of unique labeling, respectively.

Theorem 5. *The problem of assigning unique labels to n agents requires $\Omega(n \log n)$ interactions w.h.p. and the agents have to be equipped with at least n states.*

Proof. $\Omega(n \log n)$ interactions are needed w.h.p. since each agent has to interact at least once, see, e.g., the introduction in [11]. The lower bound on the number of states follows from the symmetry of agents, so any agent (different from the leader if this is given a priori) has to be prepared to be assigned an arbitrary label with at least a logarithmic bit representation. □

5.1 A sharper lower bound on the number of states

Recall that a labeling protocol is *safe* if, for any given set of agents: (i) there exists a finite run of the protocol that produces a valid agent labeling; and (ii) at any time step in any run of the protocol, no two agents have the same label.

Theorem 6. *A safe protocol for assigning unique labels to n agents requires at least $n + \sqrt{n} - 1$ states. Also, if a safe protocol uses $n + t$ states where $t < n$ then the expected number of interactions required by the protocol to achieve a valid labeling is $\frac{n^2}{t+1}$.*

Proof. Consider a finite run (i.e., a finite sequence of interactions) of the safe protocol in which each agent gets assigned a distinct label at the end. There exists such a finite run by the condition (i).

Let F be the set of final (i.e., last) states achieved by the agents at the end of the run, where distinct labels are assigned to them. We have $|F| \geq n$. Also, let R stand for the set of remaining states used in this run. Observe that if an agent is in a state in F then it has a label.

For an agent x, let $f(x) \in F$ be the last state achieved by the agent in the run, and let $pred(x)$ be the next to the last state achieved by the agent x in the run.

Next, let A be the set of agents x that achieved their final state in the run by an interaction of x in the state $pred(x)$ with an agent in a final state in F. We claim that for two distinct agents $x,\ y \in A$, $pred(x) \neq pred(y)$. Simply, otherwise there exists another run of the protocol that assigns the same final state in F and consequently the same label to both agents $x,\ y$, which contradicts condition (ii) of the definition of a safe protocol. Namely, we may assume w.l.o.g. that x gets its final state $f(x)$ in an interaction with an agent x' that already achieved its final state $f(x')$, and in a later interaction y gets its final state $f(y)$, in the original run. Then, if we replace the latter interaction by the interaction between y and the agent x' in the state $f(x')$, it will result in achieving by y the state $f(x)$ since $pred(x) = pred(y)$. We obtain a contradiction with the condition (ii) since y is in the same state in F as x and hence both have the same label at the same time step directly after reaching the state $f(x)$ by y. Hence, $|R| \geq |A|$ holds.

Let B be the set of remaining agents z that got their final state in F in an interaction where both agents have been in states outside F, i.e., in R. Since the agents in B achieved distinct final states with distinct labels in the aforementioned interactions, we infer that $|R|^2 \geq |B|$ and thus $|R| \geq \sqrt{|B|}$. (Note that if $|R|^2 < |B|$ then there would be a pair of agents in B that would achieve the same last state in the run and hence it would have the same label at the end of the considered run.)

Thus, we obtain $|R| \geq \max\{|A|, \sqrt{n - |A|}\} \geq \sqrt{n} - 1$ by straightforward calculations. This completes the proof of the first part.

To prove the second part, we may assume w.l.o.g. that $|A| < n$ since otherwise $t \geq |R| \geq |A| \geq n$. Hence, the set B of agents is non-empty. Let x be a last agent in B that being in the state $pred(x)$ gets its final state $f(x)$ by an interaction with another agent y in a state s. If y belongs to B then both x and y are the two last agents in B that simultaneously get their final states in F in the same interaction. The probability of the interaction between them is only $\frac{1}{n^2}$. Suppose in turn that y belongs to A. We know that $t \geq |R| \geq |A|$ from the previous part. Thus, there are at most t agents in B in the state s with which the agent x in the state $pred(x)$ could interact. The probability of such an interaction is at most $\frac{t}{n^2}$. We conclude that the probability of an interaction between the agent x and the agent y after which x gets its final state $f(x)$ is at most $\frac{t+1}{n^2}$ which proves the second part. □

We can also obtain an analogous lower bound on the number of states required by a silent protocol which provides a valid labeling w.h.p. The general proof idea is analogous to that of Theorem 6. However, showing the existence a finite run Z for which $|R| \geq |A|$ requires a substantial effort.

Theorem 7. *A silent protocol which assigns unique labels to n agents with probability larger than $1 - \frac{1}{n}$ requires at least $n + \sqrt{n} - 1$ states. Also, if a silent protocol provides a valid labeling with probability 1 and uses $n + t$ states where $t < n$ then the expected number of interactions required by the protocol to provide a valid labeling is $\frac{n^2}{t+1}$.*

Proof. Let I be the set of ordered pairs of the n agents. I can be interpreted as the set of possible pairwise interactions between the agents.

Let Z be a finite run of the protocol, i.e., a finite sequence of pairs in I. Suppose that after the execution of Z, each agent reaches a final state with a distinct label.

Let F_Z be the set of final states achieved by the agents after the execution of the run Z. We have $|F_Z| \geq n$. Also, let R_Z stand for the set of remaining states used in this run. Observe that if an agent is in a state in F_Z then it has a label.

For an agent x, let $f_Z(x) \in F_Z$ be the last state achieved by the agent in the run Z, and let $pred_Z(x)$ be the next to the last state achieved by the agent x in the run.

Next, let A_Z be the set of agents x that achieved their final state in the run Z by an interaction of x in the state $pred_Z(x)$ with an agent in a final state in F_Z. We claim that there is a finite run Z of the protocol such that after the execution of Z, each agent is in a final state with a distinct label and for any pair of distinct agents $x,\ y \in A_Z$, $pred(x) \neq pred(y)$.

The proof of the claim is by a contradiction. The general intuition is that if $pred_Z(x) = pred_Z(y)$ for two agents $x,\ y \in A_Z$ then we can associate with a prefix of Z a slightly modified equally likely run Z' which assigns the same label to a pair of agents.

To obtain the contradiction, we assume that for each finite run Z in which the agents achieve final states with distinct labels, there is a pair of agents $x,\ y \in A_Z$, where $pred_Z(x) = pred_Z(y)$. Let us consider such a pair of agents $x, y \in A_Z$ that minimizes the length of the prefix of Z in which both agents achieve their final stages in F_Z. We may assume w.l.o.g. that x gets its final state $f_Z(x)$ in an interaction i_1 with an agent x' that already achieved its final state $f_Z(x')$, and in a later interaction i_2, y gets its final state $f_Z(y)$, in the run Z. Thus, the shortest prefix of Z in which both x and y get their final stages has the form $Z_1 i_1 Z_2 i_2$. Then, if we replace the latter interaction i_2 by the interaction i_3 between y and the agent x' in the state $f_Z(x')$ analogous to i_1, it will result in achieving by y the state $f_Z(x)$ since $pred_Z(x) = pred_Z(y)$. Thus, neither the run $Z_1 i_1 Z_2 i_3$ nor any of its extensions yield valid labeling of the agents. Importantly, the runs $Z_1 i_1 Z_2 i_2$ and $Z_1 i_1 Z_2 i_3$ are equally likely (*).

We initialize two sets S_{valid} and $S_{invalid}$ of strings (sequences) over the alphabet I. Then, for each run Z in which the agents achieve final states with distinct labels, we insert the prefix $Z_1 i_1 Z_2 i_2$ into S_{valid} and the corresponding sequence $Z_1 i_1 Z_2 i_3$ into $S_{invalid}$. Note that by the choice of $i_1,\ i_2,$ no string in S_{valid} is a prefix of another string in S_{valid}. The analogous property holds for $S_{invalid}$. By the construction of the sets, each run Z in which the agents achieve final states with distinct labels has to overlap or be a lengthening of a string in S_{valid}. Furthermore, no run of the protocol that overlaps with a string in $S_{invalid}$ or it is a lengthening of a string in $S_{invalid}$ results in a valid labeling. Define the function $g; S_{valid} \rightarrow S_{invalid}$ by $g(Z_1 i_1 Z_2 i_2) = Z_1 i_1 Z_2 i_3$. By the property (*), the probability that a string over I is equal to $Z_1 i_1 Z_2 i_2$ or it is a lengthening of $Z_1 i_1 Z_2 i_2$ is not greater than the probability that a string over I is equal to $g(Z_1 i_1 Z_2 i_2)$ or it is a lengthening of $g(Z_1 i_1 Z_2 i_2)$. The function g is not necessarily a bijection. Suppose that $g(Z_1 i_1 Z_2 i_2) = g(Z_1' i_1' Z_2' i_2')$. Then, we have $Z_1 i_1 Z_2 i_3 = Z_1' i_1' Z_2 i_3$. Consequently, the strings $Z_1 i_1 Z_2 i_2$ and $Z_1' i_1' Z_2' i_2'$ may only differ in the last interaction, i.e., i_2 may be different from i_2'. However, i_2 and i_2' have to include the same agent (y in the earlier construction) that appears in i_3. We conclude that the aforementioned two strings in S_{valid} can differ by at most one agent in the last interaction. It follows that g maps at most $n-1$ strings in S_{valid} to the same string in $S_{invalid}$. Consequently, the event that the agents eventually achieve their final states yielding a valid labeling is at most $n-1$ times more likely than the complement event. We obtain a contradiction with theorem assumptions.

Thus, we may assume that we have a finite run Z in which the agents achieve final states with distinct labels and for any pair of agents $x,\ y \in A_Z, pred_Z(x \neq pred_Z(y)$ holds. Consequently, we have $|R_Z| \geq |A_Z|$.

The remaining part of the proof of the theorem goes exactly along the lines of the corresponding part of the proof of the first statement in Theorem 6. It is sufficient to replace $A,\ B,\ F,\ R,\ f,\ pred$ by $A_Z,\ B_Z, F_Z,\ R_Z\ f_Z,\ pred_Z$ in the aforementioned part.

Also the proof of the second statement of the theorem can be obtained from the proof of the second statement of Theorem 6 by the aforementioned replacement. For this purpose however, we need $|R_Z| \geq |A_Z|$ to hold for any run Z resulting in a valid labeling of the agents. The existence of such a run Z showed in the proof of the first statement is not sufficient to obtain a lower bound on the expected number of required iterations. The stronger assumption on the silent protocol in the second statement in the theorem requiring the protocol to provide always a valid labeling solves the problem. Simply, if $pred_Z(x) = pred_Z(y)$ for $x,\ t \in A_Z$ then neither $Z_1 i_1 Z_2 i_3$ nor any of its lengthening can provide a valid labeling. We obtain a contradiction with the aforementioned assumption. Thus, the inequality $|R_Z| \geq |A_Z|$ holds for arbitrary run Z ending with a valid labeling. □

Corollary 3. *If there exist $\varepsilon > 0$ and a safe or silent protocol which assigns unique labels to n agents (the latter with probability larger than $1 - \frac{1}{n}$) such that the protocol uses only $n + O(n^{1-\varepsilon})$ states then the expected number of interactions required by the protocol to achieve a valid labeling is $\Omega(n^{1+\varepsilon})$.*

5.2 A lower bound for the range $[1, n + r]$

Our fast protocols presented in Sections 4 are examples of a class of natural protocols for the unique labeling problem that we term *pool protocols*.

During each step of a pool protocol a subset of agents owns explicit or implicit pools of labels which are pairwise disjoint and whose union is included in the assumed range of labels. When two agents interact, they can repartition the union of their pools among themselves. Before the start of a pool protocol, only a single agent (the leader) owns a pool of labels. This initial pool corresponds to the assumed range of labels. An agent can be assigned only a label from own pool. After that the label is removed from the pool and it cannot be charged. Finally, an agent without assigned label cannot give away the whole own pool during an interaction with another agent without getting some part of the pool belonging to the other agent.

Note that a pool protocol in particular satisfies the second condition in the definition of a safe labeling protocol.

Theorem 8. *The expected number of interactions required by a pool protocol to assign unique labels in the range $[1, n + r]$, where $r \geq 0$, to the population of n agents is at least $\frac{n^2}{r+1}$.*

Proof. We shall say that an agent has the P property if the agent owns a non-empty pool or a label has been assigned to the agent. Observe that if an agent accomplishes the P property during running a pool protocol then it never loses it. Also, all agents have to accomplish the P property sooner or later in order to complete the assignment task. During each interaction of a pool protocol at most one more agent can get the P property. Since at the beginning only one agent has the P property, there must exist an interaction after which only one agent lacks this property. By the disjointedness of the pools and labels, the assumed label range, and the definition of a pool protocol, there are at most $r+1$ agents among the remaining ones that could donate a sub-pool or label from own pool to the agent missing the P property. The expected number of interactions leading to an interaction between the agent missing the P property and one of the at most $r+1$ agents is $\frac{n^2}{r+1}$. □

6 Final Remarks

Our upper bound of $n + 5 \cdot \sqrt{n} + 4$ on the number of states required by a silent and safe protocol for unique labeling almost matches our lower bound of $n + \sqrt{n} - 1$.

Our generalized fast safe protocol needs $O(n \log n/\varepsilon)$ interactions to assign unique labels from $[1, (1 + \varepsilon)n]$ to the n agents, w.h.p under the assumption that one of the agents knows n initially. It uses $(2 + \varepsilon)n + O(\log n)$ states. For a small fixed ε, the number of required interactions is asymptotically optimal, and the size of label range is close to optimal. Also, the number of states used by the protocol is close to optimal. Note that the additive term $O(\log n)$ can be shadowed by $n\varepsilon$ as long as $\varepsilon = \omega(\frac{\log n}{n})$ by the trick of decreasing ε by a constant multiplicative factor. In fact, one can modify the generalized protocol to get rid of the additive $O(\log n)$ completely. The idea is to use assigning labels to a subset of agents of logarithmic size for counting simultaneously.

We can combine our protocols for unique labeling with the recent protocols for counting or approximating the population size due to Berenbrink et al. [11] in order to get rid of the assumption that the population size is known to one of the agents initially. Since the aforementioned protocols from [11] either require $\tilde{O}(n)$ states or $O(n \log^2 n)$ interactions, the resulting combinations lose some of the near-optimality or optimality properties of our protocols (cf. Corollaries 1, 2). The related question if one can design a protocol for counting or closely approximating the population size simultaneously requiring $O(n \log n)$ interactions w.h.p. and at most cn states, where c is a low constant, is of interest in its own rights.

Acknowledgments

The authors were supported in part by Swedish Research Council grant 621-2017-03750.

References

1. D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili and R.L. Rivest. Time-Space Trade-offs in Population Protocols. *Proc. 28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pp. 2560–2579.
2. D. Alistarh, O. Denysyuk, L. Rodrigues, and N. Shavit. Balls-into-leaves: Sub-logarithmic renaming in synchronous message-passing systems. *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, pp. 232-241, 2014.
3. D. Alistarh, H. Attiya, S. Gilbert, A. Giurgiu, and R. Guerraoui. Fast randomized test-and-set and renaming. *DISC 2010*, vol. 6343, pp. 94-108.
4. D. Alistarh and R. Gelashvili. Recent Algorithmic Advances in Population Protocols. *ACM SIGACT News, October 2018* https://doi.org/10.1145/3289137.3289150.
5. D. Angluin, J. Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing, 18(4)*, pp. 235–253, 2006.
6. D. Angluin, J. Aspnes, and D. Eisenstat. Fast Computation by Population Protocols with a Leader. *Proc. International Symposium on Distributed Computing DISC 2006*, pp 61-75.
7. J. Aspnes, J. Beauquier, J. Burman, and D. Sohier. Time and space optimal counting in population protocols. *20th International Conference on Principles of Distributed Systems, OPODIS 2016*, , pp. 13:1–13:17.
8. J. Beauquier, J. Burman, L. Rosaz, and B. Rozoy. Non-deterministic Population Protocols. *Proceedings of OPODIS 2012, LNCS, Vol. 7702*, pp. 61-75, 2012.
9. P. Berenbrink, A. Brinkmann, R. Elsässer, T. Friedetzky, and L. Nagel. Randomized Renaming in Shared Memory Systems. *EEE Int. Parallel and Distributed Processing Symp. 2015*.
10. P. Berenbrink, D. Kaaser, P. Kling, and L. Otterbach. Simple and efficient leader election. *Proc. Symposium on Simplicity in Algorithms, SOSA 2018*, 9:1–9:11.
11. P. Berenbrink, D. Kaaser, T. Radzik. On Counting the Population Size. *Proc. PODC 2019*, pp. 43-52. CoRR abs/1905.11962 (2019)
12. A. Bilke, C. Cooper, R. Elsässer, and T. Radzik. Brief announcement: Population protocols for leader election and exact majority with $O(\log^2 n)$ states and $O(\log^2 n)$ convergence time. *Proc. 36th ACM Symp. on Principles of Distributed Computing, PODC 2017*, pp. 451–453.
13. P. Berenbrink, R. Elsässer, T. Friedetzky, D. Kaaser, P. Kling, and T. Radzik. Majority and stabilization in population protocols. CoRR, abs/1805.04586, 2018.
14. P. Berenbrink, G. Giakkoupis and P. Kling. Optimal time and space leader election in population protocols. *Proc. STOC 2020*, pp. 119-129.
15. J. Burman, J. Beauquier, and D. Sohier. Space-Optimal Naming in Population Protocols. *Proc. DISC 2019, LNCS, Springer* (see also Technical report, CNRS, France, 2019. (https://hal.inria.fr/hal-01790517v3).
16. J. Burman, H. Chen, H. Chen, D. Doty, T. Nowak, E. Severson, and C. Xu. Time-optimal self-stabilizing leader election in population protocols *To appear in proc. PODC 2021* (https://arxiv.org/abs/1907.06068).
17. S. Cai, T. Izumi, and K. Wada. How to Prove Impossibility Under Global Fairness: On Space Complexity of Self-Stabilizing Leader Election on a Population Protocol Model *Theory of Computing Systems*, vol. 50, pp. 433–445(2012).
18. A. Castañeda, S. Rajsbaum and M. Raynal. The renaming problem in shared memory systems: An introduction. *Comput. Sci. Rev.*, vol. 5, no. 3, pp. 229-251, Aug. 2011.
19. D. Doty and M. Eftekhari. A survey of size counting in population protocols, arXiv preprint arXiv:2105.05408, 2021.

20. R. Elsässer, T. Radzik. Recent Results in Population Protocols for Exact Majority and Leader Election. *Bull. EATCS 126 (2018)*.
21. L. Gąsieniec and G. Stachowiak. Fast space optimal leader election in population protocols. *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, 2018*, pages 2653–2667.
22. L. Gąsieniec and G. Stachowiak. Enhanced Phase Clock, Population Protocols, and Fast Space Optimal Leader Election. *JACM 68(1)*, article 2.
23. L. Gąsieniec, G. Stachowiak, and P. Uznański. Almost logarithmic-time space optimal leader election in population protocols. *Proc. SPAA'19*, pp. 93–102.