A Faster Counting Protocol for Anonymous Dynamic Networks

Alessia Milani[1] and Miguel A. Mosteiro[2]

1 **LABRI, University of Bordeaux 1, INP, Talence, France**
milani@labri.fr
2 **Department of Computer Science, Kean University, Union, USA**
mmosteir@kean.edu

Abstract

We study the problem of counting the number of nodes in a slotted-time communication network, under the challenging assumption that nodes do not have identifiers and the network topology changes frequently. That is, for each time slot links among nodes can change arbitrarily provided that the network is always connected.

This network model has been motivated by the ongoing development of new communication technologies that enable the deployment of a massive number of devices with highly dynamic connectivity patterns. Tolerating dynamic topologies is clearly crucial in face of mobility and unreliable communication. Current communication networks do have node identifiers though. Nevertheless, in future massive networks, it might be suitable to avoid nodes IDs to facilitate mass production. Consequently, knowing what is the cost of anonymity is of paramount importance to understand what is feasible or not for future generations of Dynamic Networks.

Counting is a fundamental task in distributed computing since knowing the size of the system often facilitates the desing of solutions for more complex problems. Also, the size of the system is usually used to decide termination in distributed algorithms. Currently, the best upper bound proved on the running time to compute the exact network size is double-exponential. However, only linear complexity lower bounds are known, leaving open the question of whether efficient Counting protocols for Anonymous Dynamic Networks exist or not.

In this paper we make a significant step towards answering this question by presenting a distributed Counting protocol for Anonymous Dynamic Networks which has exponential time complexity. This algorithm, which we call Incremental Counting, ensures that eventually every node knows the exact size of the system and stops executing the protocol. Previous Counting protocols have either double-exponential time complexity, or they are exponential but do not terminate, or terminate but do not provide running-time guarantees, or guarantee only an exponential upper bound on the network size. Other protocols are heuristic and do not guarantee the correct count.

1998 ACM Subject Classification F.2.0 General

Keywords and phrases Anonymous Dynamic Networks, Counting, Time-varying Graphs

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2015.28

1 Introduction

We study the problem of *Counting* the number of nodes in a communication network, under the challenging assumption that nodes do not have identifiers (IDs) and the network topology changes frequently. We consider broadcast networks in slotted-time scenarios. That is, in any given time slot, a message sent by a given node is received by all nodes directly connected to it (*one-hop neighbors*). Worst-case topology changes are modeled assuming the presence


19th International Conference on Principles of Distributed Systems (OPODIS 2015).
Editors: Emmanuelle Anceaume, Christian Cachin, and Maria Potop-Gradinariu; Article No. 28; pp. 28:1–28:13

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of an adversary that, for each time slot, chooses the set of links among nodes. The choice is arbitrary as long as, in each time slot, the network is connected. This dynamic topology model, called *1-interval connectivity*, was introduced in [10] for Dynamic Networks where each node has a unique identifier.

The network model described, called *Anonymous Dynamic Network*, has attracted a lot of attention recently [12, 4, 5, 6]. The model has been motivated by the ongoing development of new communication technologies that enable the deployment of a massive number of devices with highly dynamic connectivity patterns. Tolerating dynamic topologies is clearly crucial in face of mobility and unreliable communication. Current communication networks do have node IDs (or otherwise a labeling is defined at startup). Nevertheless, in future massive networks, it might be suitable to avoid nodes IDs to facilitate mass production. Consequently, knowing what is the cost of anonymity is of paramount importance to understand what is feasible or not for future generations of Dynamic Networks.

Counting is a fundamental distributed computing problem since knowing the size of the system facilitates the solution of more complex problems. Also this parameter is usually used to ensure the termination of the algorithm.

Counting can be solved in Anonymous Dynamic Networks, but the best known upper bound on the time complexity is double-exponential [4]. A double-exponential running time precludes the application of such algorithm to networks of significant size, but only linear lower bounds are known. Such a large gap leaves open the question of whether practical protocols exist or not.

The protocol presented in this paper makes a significant step towards answering the latter question, reducing the time complexity for *exact* Counting to exponential. Our algorithm, which we call Incremental Counting, ensures that there is a time slot when all nodes know the *exact size* of the system and they stop executing the algorithm. All nodes stop at the same round and this is known by every node. Thus it is easy to concatenate another algorithm which uses the system size.

Previous Counting protocols for Anonymous Dynamic Networks have either double-exponential time complexity [4], or they are exponential but do not terminate [4], or terminate but do not provide running-time guarantees [5], or guarantee only an exponential upper bound on the network size [12]. Other protocols are heuristic and do not guarantee the correct count [6].

All current Counting protocols for Anonymous Dynamic Networks [5, 12, 4, 6] assume the presence of one distinguished node, usually called *leader*, and additionally use some knowledge of the number of neighbors of each node, called *degree*. In our model, we include both assumptions. Namely, the presence of a leader node, and an upper bound on the maximum degree of the adversarial topology which is known by all nodes. While these assumptions may seem too strong it was proved in [12] that Counting is not solvable in Anonymous Networks without the presence of a *leader*, even if the topology does not change. In the same work, it was conjectured that any non-trivial computation is impossible without knowledge of some network characteristics.

Incremental Counting is inspired by the algorithm presented by Di Luna et al. in [4], which starts computing an upper bound on the network size using the algorithm presented in [12]. Then, it verifies each candidate size down to the correct size. To verify each candidate size, an energy-transfer approach is used. Namely, each non-leader node is initially assigned a unit of energy which is shared evenly with neighbors in each communication round, except for the leader that works as a sink. This energy-transfer protocol is a backwards version of *mass-distribution* and *gossip-based* algorithms [9, 1, 8] used to compute the size in other

network models. The unit mass initially held in only one node in the latter system is shared throughout the network, converging to the average which is the inverse of the size. The energy-transfer protocol is shown to be at most exponential in the candidate size which in turn is exponential in the worst case, yielding a double-exponential Counting protocol.

Starting with an upper bound on the size of the system (as in previous works) facilitates the verification phase, because it tells the leader after how many rounds it has "heard" (i.e. received any needed information) from all the nodes. Unfortunately, it is not known how to obtain an upper bound better than exponential, and if an upper bound is not known, the challenge is to understand which is the condition the leader has to check to know that all nodes have been heard.

In this paper, Incremental Counting leverages the above idea of verifying candidate sizes using an energy-transfer protocol, but rather than starting with an upper bound, it follows a bottom-up approach. That is, it verifies $2, 3, \ldots$, etc. up to the actual size. Then, an energy threshold is carefully chosen to decide when the count is accurate. This novel approach yields an exponential speedup in the worst-case running-time guarantees.

The running time proved also identifies the collection of energy at the leader as the speedup bottleneck for gossip-based Counting, given that all other factors in the time complexity obtained are polynomial. In contrast, in the running time of other exact Counting protocols that terminate, all factors are exponential or double exponential [4], or the running time is not proved [5].

Contributions

In the following we summarize the main contributions of our work.

- Following-up on the Conscious Counting protocol of [4], we present an improved Incremental Counting protocol for Anonymous Dynamic Networks that computes the exact number of nodes in less than $(2\Delta)^{n+1}(n+1)\ln(n+1)/\ln(2\Delta)$ communication rounds, where n is the number of nodes and Δ is any upper bound on the maximum number of neighbors that any node will ever have. Incremental Counting tolerates worst-case changes of topology, limited to 1-interval connectivity. The protocol requires the presence of one leader node and knowledge of Δ.
- The running time of Incremental Counting entails an exponential speedup over the previous best Counting algorithm in [4], which was proved to run in $O(e^{(\Delta^{2n})}\Delta^{3n})$ communication rounds, which is double-exponential. The speedup attained is mainly due to a carefully chosen energy threshold used to verify candidate sizes that are not bigger than the actual size. Our analysis shows the correctness of such verification.
- The time complexity proved identifies the phase where the leader collects energy from all other nodes as the speedup bottleneck for Counting with gossip-based protocols. Indeed, the exponential cost is due to this collection, whereas all other terms in the time complexity are polynomial. In contrast, in the running time of [4] all terms are exponential or double exponential.

Roadmap

The rest of the paper is organized as follows. In Section 2 we briefly overview previous work directly related to this paper. After formally defining the model and the problem in Section 3, we present the Incremental Counting protocol in Section 4 and its analysis in Section 5.

2 Related Work

The following is an overview of previous work on Counting in Anonymous Dynamic Networks directly related to this paper. Other related work may be found in a survey on Dynamic Networks and Time-varying Graphs by Casteigts et al. [3], and in the papers cited below.

Worst-case topology changes in Dynamic Networks may be limited assuming that the network is always connected (cf. [12, 10, 14, 4]), or sometimes disconnected but for some limited time (cf. [2, 15, 7, 13]). The *T-interval* connectivity model was introduced in [10]. For $T \geq 1$, a network is said to be T-interval connected if for every T consecutive rounds the network topology contains a stable connected subgraph spanning all nodes. In the same paper, a Counting protocol was presented, but it requires each node to have a unique identifier. In [10] it is also proved that, if no restriction on the size of the messages is required, the counting problem can be easily solved in $O(n)$ time when nodes have IDs. In our work, we focus on *Anonymous* Dynamic Networks. Understanding if a linear counting algorithm exists also when IDs are not available will help to understand the difficulty introduced by anonymity (if any).

A Counting protocol for Anonymous Dynamic Networks where an upper bound Δ on the maximum degree is known was presented in [12]. The adversarial topology is limited only to 1-interval connectivity, but the algorithm obtains only an upper bound on the size of the network n, which in the worst case is exponential, namely $O(\Delta^n)$. In our work, we aim to obtain an exact count, rather than only an upper bound.

The *Conscious Counting* algorithm presented later in [4] does obtain the exact count for the same network model, but requires knowledge of an initial upper bound K on the size of the network. Conscious Counting would be exponential if such upper bound were tight, since it runs in $O(e^{K^2}K^3)$ communication rounds. However, K is obtained using the algorithm in [12] mentioned above. Consequently, in the worst case the overall running time of the Conscious Counting Algorithm is $O(e^{(\Delta^{2n})}\Delta^{3n})$, which is double-exponential. In our work, we obtain the exact count in exponential time. That is, we reduce exponentially the best known upper bound for exact Counting.

Anonymous Dynamic Networks where an upper bound on the maximum degree is not known where also studied [4, 5, 6]. In [4], the protocol does not have a termination condition. That is, nodes running the protocol do not know whether the correct count has been reached or not. Hence, they have to continue running the protocol forever. In a companion paper [6], the authors stop the protocol heuristically. Hence, the count obtained is not guaranteed to be correct. Indeed, errors appear when the conductance of the underlying connectivity graph is low. In our work, we aim for Counting algorithms that terminate returning always the correct count. The protocol in [5] is shown to eventually terminate, although the running time is not proved. In their model, it is assumed that each node is equipped with an oracle that provides an estimation of its degree at each round. This is still an assumption of knowledge of network characteristics, although local. This and the above shortcomings are not unexpected in light of the conjecture in [12], which states that Counting (actually, any non-trivial computations) in Anonymous Dynamic Networks without knowledge of some network characteristics is impossible. Nevertheless, a proof of such conjecture has not been found yet.

Known lower bounds for Counting in Anonymous Dynamic Networks include only the trivial $\Omega(D)$, where D is the *dynamic* diameter of the network, and $\Omega(\log n)$ [1] even if D is constant, proved in [11].

[1] Throughout the paper, log means logarithm base 2, unless otherwise stated.

3 Preliminaries

3.1 The Counting Problem

An algorithm is said to solve the *Counting* problem if whenever it is executed in a Dynamic Network comprising n nodes, all nodes eventually terminate and output n.

3.2 The Anonymous Dynamic Network Model

We consider a synchronous Dynamic Network composed of a fixed set of nodes V where $|V| = n$. Nodes have no identifiers (IDs) or labels. We also assume the presence of a special node called the *leader* and denoted ℓ.

Nodes communicate by broadcast. In particular, communication proceeds in synchronous *rounds*. At each round a node broadcasts a message to its neighbors and simultaneously receives the messages broadcast in the same round by its neighbors (if any), then it makes some local computation. The time of computation is negligible. Thus, we compute the time complexity in rounds of communication.

At each round the set of communication links changes adversarially. Thus, the network is modeled as a dynamic graph $G = (V, E)$ where $E : \mathbb{N} \to \{(u, v) s.t. (u, v) \in V\}$ is a function mapping a round number r to a set of undirected edges $E(r)$. In particular, we consider the following 1-interval connectivity model proposed by Kuhn et al. in [10].

▶ **Definition 1.** A dynamic graph $G = (V, E)$ is 1-interval connected if for all $r \in \mathbb{N}$, the static graph $G_r := (V, E(r))$ is connected.

Finally, we assume that the size of the neighborhood of a node is upper bounded by a number $\Delta > 0$ at every round, and we assume that Δ is known by the nodes.

At a first glance, some knowledge of the degree seems unnecessary because, after one message from each neighbor has been received in a given round, the degree is simply the message count. However, for the next round of communication, the degree may change due to changing topology. Thus, a node does not know its current degree before sending messages to its neighbors.

4 Distributed Counting Algorithm

Incremental Counting consists of a sequence of iterations. In each iteration, a candidate size is checked to decide if it is correct. If not, the candidate size is increased and a new iteration starts. In the following, we provide a high level explanation of the algorithm executed in each iteration.

At the beginning of each iteration every node is assigned energy value 1, except for the leader which has 0 energy. Then, the iteration proceeds in three consecutive phases described below. Each phase lasts a fixed amount of rounds which only depends on the current estimation of the system size. This is intended to synchronize the computation at all the nodes in the system without extra communication.

During the first phase, called the *Collection Phase*, each node discharges itself by sending at each round a fraction at most half of its current energy to its neighbors. Then it computes its new energy by taking into account the energy given to its neighbors and the energy received from them. The leader acts as a sink collecting energy but not disseminating it. This phase completes when the leader has received an amount of energy such that, if the candidate size for the current iteration is the correct system size n, there is no node in the

system with more than $1/k^c$ residual energy, for some constant $c > 1$. The function $\tau(k)$ in Algorithms 1 and 2 gives the number of iterations of the Collection Phase needed to guarantee this. An exponential upper bound on $\tau(k)$ is computed in Corollary 7. However, the bound may not be tight, so $\tau(k)$ is left as a parameter in the protocol. Should a better bound on $\tau(k)$ be proved, the protocol can be used as is.

Then, the *Verification Phase* starts. During this phase, the energy at each node does not change and the leader verifies the correctness of the current candidate size looking for a node with residual energy greater than $1/k^c$. To this aim at each round of the Verification phase each non leader node broadcasts the maximal energy it has "heard" during this phase. At the beginning each such node broadcasts its own residual energy. This phase lasts sufficiently long to ensure that if a node with residual energy greater than $1/k^c$ exists, then the leader will hear from it. If the leader does not hear from such node, it knows that the candidate size was indeed correct, and the verification phase completes successfully.

The last phase, called *Notification Phase*, is used by the leader when the verification phase completes successfully. To notify such event, the leader broadcasts a special $\langle Halt \rangle$ message, and each node in turn broadcasts it as soon as it is received and as long as the Notification Phase is not completed. If the Verification Phase completes unsuccessfully, the leader and every other node simply wait for the same number of rounds of communication without taking any action, and then all the nodes start a new iteration. This procedure ensures synchronism. A node stops executing the algorithm at the end of the Notification phase if it has received the $\langle Halt \rangle$ message. At this time every node knows the exact size of the system.

The Incremental Counting protocol for the leader and non-leader nodes is detailed in Algorithms 1 and 2.

PseudoCode

Variables at the leader node

- e_ℓ is the energy of the leader at the current round. It is initialized to 0 at the beginning of each iteration.
- k is the estimation of the system size. Initially equal to 1 and increased by one in each iteration.
- $1/k^c$ is a threshold value for the energy such that, for a given estimate k, if k is the correct size of the system, after the Collection Phase no node has energy greater than $1/k^c$ for some constant $c > 1$.
- $IsCorrect$, initially $true$ is set to $false$ if the leader discovers that its estimate k is wrong. This happens if the value of $e_\ell > k - 1$ at the end of the Collection phase or if during the Verification phase the leader discovers a node with energy greater than $1/k^c$.
- $halt$, initially $false$ is set to $true$ when the leader verifies that k is the correct size of the system.

Variables at non leader nodes

- e is the energy of the node at the current round. It is initialized to 1 at the beginning of each iteration.
- k is the estimation of the system size. Initially equal to 1 and increased by one in each iteration.
- e_{max}, is the maximum energy the node is aware of at the current round of the Verification Phase.
- $halt$, initially $false$, is set to $true$ when the node receives a $\langle Halt \rangle$ message.

Algorithm 1: INCREMENTAL COUNTING algorithm for the leader node.

```
1  k ← 1
2  halt ← false
3  while ¬halt do
4      k ← k + 1
5      IsCorrect ← true
6      e_ℓ ← 0
       // Collection Phase
7      for each of τ(k) communication rounds do
8          receive e_1, e_2, ... e_s from neighbors, where 1 ≤ s ≤ Δ
9          e_ℓ ← e_ℓ + e_1 + e_2 + ... + e_s
       // Verification Phase
10     for each of 1 + ⌈k/(1−1/k^c)⌉ communication rounds do
11         receive e_1, e_2, ... e_s from neighbors, where 1 ≤ s ≤ Δ
12         if k − 1 − 1/k^c ≤ e_ℓ ≤ k − 1 then
13             for j := 1 ... s do
14                 if e_j > 1/k^c then
15                     IsCorrect ← false
16         else
17             IsCorrect ← false
       // Notification Phase
18     for each of k communication rounds do
19         if IsCorrect then
20             broadcast ⟨Halt⟩
21             halt ← true
22         else
23             do nothing
24 output k
```

5 Analysis

The following notation will be used. The energy of node i at the beginning of round r, is denoted as e_i^r, which is also generalized to any set of nodes $S \subseteq V$ as $e_S^r = \sum_{i \in S} e_i^r$. For any given round r and node i, let the set of neighbors of i be N_i^r and the average energy of i's neighbors be $\bar{e}_{N_i^r}$. The superindex indicating the round number will be omitted when clear from context or irrelevant. Also, at any time, let $\sum_{i \in V} e_i$ be called the *system energy* and $\sum_{i \in V \setminus \{\ell\}} e_i$ be called the *energy left*. At the beginning of each iteration of the protocol, that is, for each new size estimate k, the energy of the leader is reset to zero and the energy of the non-leader nodes is reset to 1. Thus, the system energy is $\sum_{i \in V} e_i = n - 1$ and the energy left is $\sum_{i \in V \setminus \{\ell\}} e_i = n - 1$.

▶ **Lemma 2.** *For any network of n nodes, including a leader ℓ, running the* INCREMENTAL COUNTING *Protocol under the communication and connectivity models defined the following holds. For any given node $i \in V \setminus \{\ell\}$ and for any given round r of the Collection Phase, it is $e_i^r \leq 1$.*

Algorithm 2: INCREMENTAL COUNTING algorithm for non-leader nodes.

```
1  k ← 1
2  halt ← false
3  while ¬halt do
4      k ← k + 1
5      e ← 1
       // Collection Phase
6      for each of τ(k) communication rounds do
7          broadcast ⟨e/(2Δ)⟩ and receive e_1, e_2, ... e_s from neighbors, where 1 ≤ s ≤ Δ
8          e ← e · (1 − s/(2Δ)) + Σ_{j=1}^{s} e_j
       // Verification Phase
9      e_max ← e
10     for each of ⌈1 + k/(1−1/k^c)⌉ communication rounds do
11         broadcast ⟨e_max⟩ and receive e_1, e_2, ... e_s from neighbors, where 1 ≤ s ≤ Δ
12         for j := 1 ... s do
13             if e_j > e_max then
14                 e_max ← e_j
       // Notification Phase
15     for each of k communication rounds do
16         if halt then
17             broadcast ⟨Halt⟩
18         if receive ⟨Halt⟩ from some neighbor
19          then
20             halt ← true
21 output k
```

Proof. Fix some arbitrary (non-leader) node i. Consider the transition between round r and $r+1$. We have that

$$e_i^{r+1} \leq e_i^r + \overline{e}_{N_i^r}\frac{|N_i^r|}{2\Delta} - e_i^r\frac{|N_i^r|}{2\Delta} = e_i^r + (\overline{e}_{N_i^r} - e_i^r)\frac{|N_i^r|}{2\Delta}.$$

If $\overline{e}_{N_i^r} \leq e_i^r$, then $e_i^{r+1} \leq e_i^r$. That is, i's energy does not increase from round r to round $r+1$. If on the other hand it is $\overline{e}_{N_i^r} > e_i^r$, we have

$$e_i^{r+1} \leq e_i^r + (\overline{e}_{N_i^r} - e_i^r)/2 = (e_i^r + \overline{e}_{N_i^r})/2.$$

That is, the energy of i in round $r+1$ is at most the average between the energy of i in round r and the average of i's neighbors' energy in round r.

Now consider the evolution of the protocol along many rounds. We ignore the rounds when $\overline{e}_{N_i^r} \leq e_i^r$ since they do not increase the energy. For the other rounds, given that all nodes start with energy 1, and that the average of some numbers cannot be bigger than the maximum, the energy at any given node cannot get bigger than 1. Hence, the claim follows. ◀

▶ **Lemma 3.** *For any network of n nodes, under the communication and connectivity models defined, the following holds. If a message m is held by all nodes in a set $V_1 \subseteq V$, after*

$|V| - |V_1|$ *rounds when every node holding the message broadcasts* m *in each round, all nodes in* V *hold the message.*

Proof. For any round $r > 0$, consider the partition of nodes $\{V_1^r, V_2^r\}$ defined by the nodes holding the message at the beginning of round r. That is, $\forall i \in V_1^r$ the node i holds m and $\forall j \in V_2^r$ the node j does not hold m. By 1-interval connectivity, there must exist a link u, v, such that $u \in V_1^r$ and $v \in V_2^r$. Given that all nodes holding the message broadcast m, v must receive the message in round r. Thus, at the beginning of round $r+1$ it is $|V_1^{r+1}| \geq |V_1^r| + 1$ and $|V_2^{r+1}| \leq |V_2^r| - 1$. Applying the same argument inductively, after $|V_2^{r+1}|$ more rounds all nodes hold the message. ◀

The following lemma is a straightforward application of Lemma 3 to the Notification Phase, where the message broadcasted is $\langle Halt \rangle$ for the first time when $k = n$.

▶ **Lemma 4** (Correctness of the Notification Phase)**.** *For any network of* n *nodes, including a leader* ℓ*, running the* Incremental Counting *Protocol under the communication and connectivity models defined the following holds. If at the end of the Verification Phase* $IsCorrect = true$*, then at the end of the Notification Phase all nodes stop the Counting Protocol holding the size* n*.*

▶ **Lemma 5** (Correctness of the Verification Phase)**.** *For any network of* $n > 3$ *nodes, including a leader* ℓ*, running the* Incremental Counting *Protocol under the communication and connectivity models defined the following holds. For any estimate of the size of the network* k *and constant* $c > 1$*, at the end of the Verification Phase* $IsCorrect = true$ *if and only if* $k = n$*.*

Proof. We start observing that, for each estimate k, each non-leader node is initialized with one unit of energy (Line 5 in Algorithm 2) and the leader's energy is initialized to 0 (Line 6 in Algorithm 1). Until a new iteration of the outer loop (in both algorithms) is executed, no energy is lost or gained by the system as a whole. Hence, the system energy is always $n - 1$.

We prove first that, if $k = n$, at the end of the Verification Phase it is $IsCorrect = true$. Given that $k = n$, the system energy is $k-1$ and therefore $e_\ell \leq k-1$. Also because $k = n$, we know that after the Collection Phase it is $e_\ell \geq k - 1 - 1/k^c$ by definition of $\tau(k)$. Therefore, $IsCorrect$ is not set to false in Line 17 of Algorithm 1. Also because $e_\ell \geq k-1-1/k^c$ at the end of the Collection Phase, we know that the energy left at the beginning of the Verification Phase is $e_{V\setminus\{\ell\}} = k - 1 - e_\ell \leq 1/k^c$. Therefore, no non-leader node could have more than that energy. That is, $\forall i \in V \setminus \{\ell\} : e_i \leq 1/k^c$. Thus, during the Verification Phase, the leader will not be able to detect a node with energy bigger than $1/k^c$. Therefore, $IsCorrect$ is not set to false in Line 15 of Algorithm 1 either. There is no other line where $IsCorrect$ is set to false. Hence, at the end of the Verification Phase it is $IsCorrect = true$.

We prove now the other direction of the implication. That is, if at the end of the Verification Phase $IsCorrect = true$, then it is $k = n$. For the sake of contradiction, assume that $IsCorrect = true$ but $k \neq n$. Notice that k cannot be larger than n, because the estimate is increased one by one, we already proved that if $k = n$ at the end of the Verification Phase it is $IsCorrect = true$, and Lemma 4 shows that all nodes would have stopped running the protocol. Thus, we are left with the case when $k < n$.

Notice that if $e_\ell > k - 1$ the variable $IsCorrect$ is set to false in Line 17 of Algorithm 1. Hence, it must be $e_\ell \leq k - 1$ and, given that the system energy is $n - 1$, the energy left is $e_{V\setminus\{\ell\}} \geq n - k$. This energy left is stored in the $n - 1$ non-leader nodes. Hence, there must exist some node $j \in V\setminus\{\ell\}$ in the network such that $e_j \geq (n-k)/(n-1)$. If $IsCorrect = true$ it means that the leader did not detect a node with energy bigger than $1/k^c$ in Line 14 of

Algorithm 1. However, for any $2 \leq k \leq n-1$, $n > 3$, and $c > 1$, it is $1/k^c < (n-k)/(n-1)$ which means that such node must exist.

To see why the latter inequality is true, we verify that $k^c(n-k) - n + 1 > 0$ as follows. With respect to k, this function has a maximum for $k = cn/(c+1)$. That is, for $2 \leq k \leq n-1$ (recall that we are in the case $k < n$), the function has minima in 2 and $n-1$. Then, it is enough to verify that $2^c(n-2) - n + 1 > 0$, which is true for any $c > 1$ and $n > 3$, and that $(n-1)^c - n + 1 > 0$, which is also true for any $c > 1$ and $n \geq 2$.

Thus, to complete the proof, it is enough to show that $1 + k^{c+1}/(k^c - 1)$ rounds are enough to detect a node with energy bigger than $1/k^c$. To do that, we upper bound the number of nodes in the network with energy at most $1/k^c$ as follows. We know that at any time when the leader has energy e_ℓ, the energy left is $n - 1 - e_\ell$. Let $S \subseteq V$ be the set of nodes with energy at most $1/k^c$. Then, we have that $n - 1 - e_\ell = \sum_{j \in S} e_j + \sum_{k \in V \setminus S} e_k$. To maximize the size of S, we minimize the size of $V \setminus S$ assuming that all nodes in $V \setminus S$ have maximum energy, which according to Lemma 2 is at most 1. Then, we have that $n - 1 - e_\ell = \sum_{j \in S} e_j + (n - |S|)$ which yields $|S| - 1 - e_\ell = \sum_{j \in S} e_j$ Given that $\sum_{j \in S} e_j \leq |S|/k^c$, we have that $|S| \leq (1 + e_\ell)/(1 - 1/k^c)$. Recall that $e_\ell \leq k - 1$ because *IsCorrect* would have been set to false in Line 17 of Algorithm 1 otherwise. Replacing, we get $|S| \leq k^{c+1}/(k^c - 1)$.

Let $\{V_1, V_2\}$ be a partition of V such that $V_2 = S \cup \{\ell\}$. Recall that, for any $v \in V_1$ it is $e_v > 1/k^c$. Using Lemma 3, we know that $|V_2| = 1 + k^{c+1}/(k^c - 1)$ iterations in the Verification Phase of Algorithm 1 are enough for the leader to detect that there is a node with energy larger than $1/k^c$, which contradicts our assumption that $IsCorrect = true$. ◀

The following theorem establishes our main result.

▶ **Theorem 6.** *For any anonymous dynamic network of $n > 3$ nodes, including a leader ℓ, and for any constant $c > \log 5$, the following holds. If the adversarial topology is limited by a maximum degree Δ and the connectivity model defined, and nodes run the* Incremental Counting *Protocol in Algorithms 1 and 2 under the communication model defined, after r rounds, all nodes stop holding the size of the network n, where*

$$r < n(n+3) + \ln n - 4 + \sum_{k=2}^{n} \tau(k).$$

Where $\tau(k)$ is a function such that, if $k = n$ and the Collection Phase is executed for at least $\tau(k)$ rounds, then at the end of the phase the leader has energy $e_\ell \geq k - 1 - 1/k^c$.

Proof. Correctness is a direct consequence of Lemmas 4 and 5. The running time is obtained adding the number of iterations of each phase, as follows.

$$\begin{aligned} r &= \sum_{k=2}^{n} \left(\tau(k) + \left\lceil 1 + \frac{k}{1 - 1/k^c} \right\rceil + k \right) \\ &\leq \sum_{k=2}^{n} \left(\tau(k) + 2 + \frac{k}{1 - 1/k^c} + k \right) \\ &= n(n+3) - 4 + \sum_{k=2}^{n} \left(\tau(k) + \frac{k}{k^c - 1} \right). \end{aligned}$$

Using that $k/(k^c-1) < 1/k$ for any $c > \log 5$ and $k \geq 2$, we obtain the following.

$$\begin{aligned} r &< n(n+3) - 4 + \sum_{k=2}^{n} \left(\tau(k) + \frac{1}{k} \right) \\ &\leq n(n+3) + \ln n - 4 + \sum_{k=2}^{n} \tau(k). \end{aligned}$$

◀

Bounding the running time of the Collection Phase using Lemma 2 in [4] in Theorem 6, the following corollary is obtained.

▶ **Corollary 7.** *For any anonymous dynamic network of $n > 6$ nodes, including a leader ℓ, the following holds. If the adversarial topology is limited by a maximum degree $1 \leq \Delta \leq n-1$ and the connectivity model defined, and nodes run the* Incremental Counting *Protocol in Algorithms 1 and 2 under the communication model defined, after r rounds, all nodes stop holding the size of the network n, where*

$$r < \frac{(2\Delta)^{n+1}(n+1)\ln(n+1)}{\ln(2\Delta)}.$$

Proof. Lemma 2 in [4] proves that, for any estimate $k \geq n$ and integer $\rho > 0$, starting with $e_\ell = 0$ and $e_i = 1$ for all $i \in V \setminus \{\ell\}$, after running ρk rounds of the energy transfer protocol the energy stored in the leader is $e_\ell \geq n(1 - (((2\Delta)^k - 1)/(2\Delta)^k)^\rho)$. Notice in Theorem 6 that the condition $e_\ell \geq k - 1 - 1/k^c$ only applies when $k = n$. Thus, it is enough to find ρ such that

$$\begin{aligned} k\left(1 - \left(\frac{(2\Delta)^k - 1}{(2\Delta)^k}\right)^\rho\right) &\geq k - 1 - 1/k^c \\ \rho &\geq \frac{\ln(k/(1+1/k^c))}{\ln\left(1/(1-1/(2\Delta)^k)\right)}. \end{aligned}$$

Using that $1 - x \leq e^{-x}$ for $x \leq 1$, it is enough to have $\rho = \lceil (2\Delta)^k \ln k \rceil$. Replacing in Theorem 6, we obtain

$$\begin{aligned} r &< n(n+3) + \ln n - 4 + \sum_{k=2}^{n} k\lceil (2\Delta)^k \ln k \rceil \\ &\leq n(n+3) + \ln n - 4 + \sum_{k=2}^{n} k(1 + (2\Delta)^k \ln k) \\ &= n(3n+7)/2 + \ln n - 5 + \sum_{k=2}^{n} k(2\Delta)^k \ln k. \end{aligned}$$

Bounding with the integral,

$$\begin{aligned} r &< n(3n+7)/2 + \ln n - 5 + \int_{k=2}^{n+1} k(2\Delta)^k \ln k \, \mathrm{d}k \\ &= n(3n+7)/2 + \ln n - 5 + \frac{(2\Delta)^k((k\ln(2\Delta) - 1)\ln k - 1) + \mathrm{Ei}(k\ln(2\Delta))}{\ln^2(2\Delta)}\Bigg|_2^{n+1} \\ &\leq n(3n+7)/2 + \ln n + \\ &\qquad \frac{(2\Delta)^{n+1}(((n+1)\ln(2\Delta) - 1)\ln(n+1) - 1) + \mathrm{Ei}((n+1)\ln(2\Delta))}{\ln^2(2\Delta)}. \end{aligned}$$

Using that $\mathrm{Ei}(\ln x) = \mathrm{li}(x) < x$, for any real number $x \neq 1$, it is $\mathrm{Ei}((n+1)\ln(2\Delta)) < (2\Delta)^{n+1}$. Replacing,

$$r < n(3n+7)/2 + \ln n + \frac{(2\Delta)^{n+1}((n+1)\ln(2\Delta) - 1)\ln(n+1)}{\ln^2(2\Delta)}$$
$$= n(3n+7)/2 + \ln n + \frac{(2\Delta)^{n+1}(n+1)\ln(n+1)}{\ln(2\Delta)} - \frac{(2\Delta)^{n+1}\ln(n+1)}{\ln^2(2\Delta)}.$$

Using that $n(3n+7)/2 + \ln n < (2\Delta)^{n+1}\ln(n+1)/\ln^2(2\Delta)$ for any $n > 6$ and $1 \leq \Delta \leq n-1$, the claim follows. ◀

5.1 Discussion

In this paper we have studied the problem of Counting in Anonymous Dynamic Networks. The problem is challenging because the lack of identifiers and changing topology make difficult to decide if a new message has been received before from the same node. Also, the obvious lack of knowledge of the network size makes difficult to decide when the algorithm has to stop.

Assuming an upper bound on the size of the system facilitates termination but may lead to very bad time complexity if the upper bound is a huge overestimate. According to our knowledge, the algorithm in [12] is the only one to compute an upper bound of the system size for Anonymous Dynamic Networks and in the worst case it is exponential, i.e. $O(\Delta^n)$ where n is the size of the system and Δ is an upper bound on the nodes' degree. Finding the termination condition when an upper bound on the network size is not available is more challenging, but it is expected to provide more efficient algorithms. Our Incremental Counting algorithm does not assume such upper bound, and computes the exact size of the system applying a bottom-up approach where the size is possibly underestimated several times.

It is known that if no restriction on the size of the messages is required, the Counting problem can be easily solved in $O(n)$ time when nodes have IDs [10]). In this paper, we have made a significant step towards understanding if a linear Counting algorithm exists also when IDs are not available, by identifying the speedup bottleneck and reducing exponentially the best known upper bound. This will help to understand the difficulty introduced by anonymity (if any). Despite our contribution, there is still a big gap with respect to the linear lower bound trivially given by the dynamic diameter.

Finally, although we focus on communication networks, our results carry over into any distributed system of similar characteristics.

Acknowledgements. We thank Arnaud Casteigts for introducing the model to us, and Antonio Fernández Anta for useful discussions.

References

1 Paulo Sérgio Almeida, Carlos Baquero, Martín Farach-Colton, Paulo Jesus, and Miguel A. Mosteiro. Fault-tolerant aggregation: Flow-updating meets mass-distribution. In Antonio Fernández Anta, Giuseppe Lipari, and Matthieu Roy, editors, *Principles of Distributed Systems – 15th International Conference, OPODIS 2011, Toulouse, France, December 13-16, 2011. Proceedings*, volume 7109 of *Lecture Notes in Computer Science*, pages 513–527. Springer, 2011. doi:10.1007/978-3-642-25873-2_35.

2 Antonio Fernández Anta, Alessia Milani, Miguel A. Mosteiro, and Shmuel Zaks. Opportunistic information dissemination in mobile ad-hoc networks: the profit of global synchrony. *Distributed Computing*, 25(4):279–296, 2012. doi:10.1007/s00446-012-0165-9.

3 Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

4 Giuseppe Antonio Di Luna, Roberto Baldoni, Silvia Bonomi, and Ioannis Chatzigiannakis. Conscious and unconscious counting on anonymous dynamic networks. In Mainak Chatterjee, Jian-nong Cao, Kishore Kothapalli, and Sergio Rajsbaum, editors, *Distributed Computing and Networking*, volume 8314 of *Lecture Notes in Computer Science*, pages 257–271. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-45249-9_17.

5 Giuseppe Antonio Di Luna, Roberto Baldoni, Silvia Bonomi, and Ioannis Chatzigiannakis. Counting in anonymous dynamic networks under worst-case adversary. In *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*, pages 338–347. IEEE, 2014.

6 Giuseppe Antonio Di Luna, Silvia Bonomi, Ioannis Chatzigiannakis, and Roberto Baldoni. Counting in anonymous dynamic networks: An experimental perspective. In Paola Flocchini, Jie Gao, Evangelos Kranakis, and Friedhelm Meyer auf der Heide, editors, *Algorithms for Sensor Systems*, volume 8243 of *Lecture Notes in Computer Science*, pages 139–154. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-45346-5_11.

7 K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, pages 27–34, 2003.

8 Antonio Fernández Anta, Miguel A. Mosteiro, and Christopher Thraves. An early-stopping protocol for computing aggregate functions in sensor networks. *J. Parallel Distrib. Comput.*, 73(2):111–121, 2013. doi:10.1016/j.jpdc.2012.09.013.

9 D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. of the 44th IEEE Ann. Symp. on Foundations of Computer Science*, pages 482–491, 2003.

10 Fabian Kuhn, Nancy Lynch, and Rotem Oshman. Distributed computation in dynamic networks. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC'10, pages 513–522, New York, NY, USA, 2010. ACM. doi:10.1145/1806689.1806760.

11 Giuseppe Antonio Di Luna and Roberto Baldoni. Investigating the cost of anonymity on dynamic networks. *CoRR*, abs/1505.03509, 2015. URL: http://arxiv.org/abs/1505.03509.

12 Othon Michail, Ioannis Chatzigiannakis, and Paul G Spirakis. Naming and counting in anonymous unknown dynamic networks. In *Stabilization, Safety, and Security of Distributed Systems*, pages 281–295. Springer, 2013.

13 Othon Michail, Ioannis Chatzigiannakis, and Paul G Spirakis. Causality, influence, and computation in possibly disconnected synchronous dynamic networks. *Journal of Parallel and Distributed Computing*, 74(1):2016–2026, 2014.

14 Regina O'Dell and Rogert Wattenhofer. Information dissemination in highly dynamic graphs. In *Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing*, DIALM-POMC'05, pages 104–110, New York, NY, USA, 2005. ACM. doi:10.1145/1080810.1080828.

15 L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.