

Brief Announcement: Naming and Counting in Anonymous Unknown Dynamic Networks*

Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis

Computer Technology Institute & Press “Diophantus” (CTI), Patras, Greece
Email: {michailo, ichtatz, spirakis}@cti.gr

Contribution. We study the fundamental naming and counting problems in networks that are anonymous, unknown, and possibly dynamic. Network dynamicity is modeled by the 1-interval connectivity model [KLO10]. We first prove that on static networks with broadcast counting is impossible to solve without a leader and that naming is impossible to solve even with a leader and even if nodes know n . These impossibilities carry over to dynamic networks as well. With a leader we solve counting in linear time. Then we focus on dynamic networks with broadcast. We show that if nodes know an upper bound on the maximum degree that will ever appear then they can obtain an upper bound on n . Finally, we replace broadcast with *one-to-each*, in which a node may send a different message to each of its neighbors. This variation is then proved to be computationally equivalent to a full-knowledge model with unique names.

The Model. A *dynamic network* is modeled by a *dynamic graph* $G = (V, E)$, where V is a static set of n nodes and $E : \mathbb{N}_{\geq 1} \rightarrow \mathcal{P}(\{\{u, v\} : u, v \in V\})$ is a function mapping a round number $r \in \mathbb{N}_{\geq 1}$ to a set $E(r)$ of bidirectional links. A dynamic graph/network $G = (V, E)$ is said to be *1-interval connected*, if, for all rounds $r \in \mathbb{N}_{\geq 1}$, the static graph $G(r)$ is connected [KLO10]. Note that this allows the connected network to change arbitrarily from round to round.

Nodes are *anonymous*, that is they do not initially have any ids, and they do not know the topology or the size of the network, apart from some minimal knowledge when necessary. However, nodes have unlimited local storage. Communication is *synchronous message passing*. We focus on the *one-to-each* message transmission model in which, in every round r , each node u generates a different message $m_{u,v}(r)$ to be delivered to each current neighbor v .

Naming Protocols. We first present a terminating protocol that assigns unique (consecutive if needed) ids to the nodes and informs them of n in $O(n)$ -time and then refine the size of its messages. We assume that there is a unique leader l with id 0 (as without it naming is impossible) while all other nodes have id \perp .

Main Idea: Named nodes assign uids and inform the leader of these assignments. At some round r , the leader knows a set of assigned ids $K(r)$. In $|K(r)|$ additional rounds the leader will hear from a node outside $K(r)$ iff $|K(r)| \neq |V|$.

Protocol *Dynamic_Naming*. Initially, every node has three variables $count \leftarrow 0$, $acks \leftarrow \emptyset$, and $latest_unassigned \leftarrow 0$ and the leader additionally has

* Supported in part by the EU (ESF) and Greek national funds, project FOCUS.....

$latest_new \leftarrow 0$, $time_bound \leftarrow 1$, and $known_ids \leftarrow \{0\}$. A node with $id \neq \perp$ for $1 \leq i \leq k$ sends $assign(id, count + i)$ message to its i th neighbor and sets $count \leftarrow count + k$. In the first round, the leader additionally sets $known_ids \leftarrow \{0, (0, 1), (0, 2), \dots, (0, k)\}$, $latest_new \leftarrow 1$, and $time_bound \leftarrow 1 + |known_ids|$. Upon receipt of l $assign$ messages (rid_j), a node with $id = \perp$ sets $id \leftarrow \min_j \{rid_j\}$ (in number of bits), $acks \leftarrow acks \cup id$, sends an $ack(acks)$ message to all its k current neighbors, for $1 \leq i \leq k$ sends $assign(id, count + i)$ message to its i th neighbor, and sets $count \leftarrow count + k$. Upon receipt of l ack messages ($acks_j$), a nonleader sets $acks \leftarrow acks \cup (\bigcup_j acks_j)$ and sends $ack(acks)$. A node with $id = \perp$ sends $unassigned(current_round)$. Upon receipt of $l \geq 0$ $unassigned$ messages (val_j), a node with $id \notin \{0, \perp\}$ sets $latest_unassigned \leftarrow \max\{latest_unassigned, \max_j \{val_j\}\}$ and sends $unassigned(latest_unassigned)$. Upon receipt of l ack messages ($acks_j$), the leader if $(\bigcup_j acks_j) \setminus known_ids \neq \emptyset$ sets $known_ids \leftarrow known_ids \cup (\bigcup_j acks_j)$, $latest_new \leftarrow current_round$ and $time_bound \leftarrow current_round + |known_ids|$ and upon receipt of l $unassigned$ messages (val_j), it sets $latest_unassigned \leftarrow \max\{latest_unassigned, \max_j \{val_j\}\}$. If, at some round r , it holds at the leader that $r > time_bound$ and $latest_unassigned < latest_new$, the leader sends a $halt(|known_ids|)$ message for $|known_ids| - 1$ rounds and then outputs id and halts. Any node that receives a $halt(n)$ message, sends $halt(n)$ for $n - 2$ rounds and then outputs id and halts.....

.....
 A drawback of *Dynamic_Naming* is its $\Theta(n^2)$ bits/message. We now refine it to reduce the message size to $\Theta(\log n)$ paying in $O(n^3)$ termination-time.

Protocol *Individual_Conversations* [Main Idea]. To reduce the size of the messages (i) the assigned names are now of the form $k \cdot d + id$, where id is the id of the node, d is the number of *unique consecutive* ids that the leader knows so far, and $k \geq 1$ is a name counter (ii) Any time that the leader wants to communicate to a remote node that has obtained a unique id it sends a message with the id of that node and a timestamp equal to the current round. The timestamp allows all nodes to prefer this message from previous ones so that the gain is twofold: the message is delivered and no node ever issues a message containing more than one id. The remote node then can reply in the same way. For the assignment formula to work, nodes that obtain ids are not allowed to further assign ids until the leader freezes all named nodes and reassigns to them unique consecutive ids. During freezing, the leader is informed of any new assignments by the named nodes and terminates if all report that no further assignments were performed. The full version is available at <http://arxiv.org/abs/1208.0180> [MCS12].

References

- [KLO10] F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 513–522. ACM, 2010.
- [MCS12] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Naming and counting in anonymous unknown dynamic networks. *CoRR*, abs/1208.0180, 2012.