# Proof-Labeling Schemes: Broadcast, Unicast and in Between

Boaz Patt-Shamir and Mor Perry(✉)

School of Electrical Engineering, Tel Aviv University, 6997801 Tel Aviv, Israel
mor@eng.tau.ac.il

**Abstract.** We study the effect of limiting the number of different messages a node can transmit simultaneously on the verification complexity of proof-labeling schemes (PLS). In a PLS, each node is given a label, and the goal is to verify, by exchanging messages over each link in each direction, that a certain global predicate is satisfied by the system configuration. We consider a single parameter $r$ that bounds the number of distinct messages that can be sent concurrently by any node: in the case $r = 1$, each node may only send the same message to all its neighbors (the broadcast model), in the case $r \geq \Delta$, where $\Delta$ is the largest node degree in the system, each neighbor may be sent a distinct message (the unicast model), and in general, for $1 \leq r \leq \Delta$, each of the $r$ messages is destined to a subset of the neighbors.

We show that message compression linear in $r$ is possible for verifying fundamental problems such as the agreement between edge endpoints on the edge state. Some problems, including verification of maximal matching, exhibit a large gap in complexity between $r = 1$ and $r > 1$. For some other important predicates, the verification complexity is insensitive to $r$, e.g., the question whether a subset of edges constitutes a spanning-tree. We also consider the congested clique model. We show that the crossing technique [5] for proving lower bounds on the verification complexity can be applied in the case of congested clique only if $r = 1$. Together with a new upper bound, this allows us to determine the verification complexity of MST in the broadcast clique.

**Keywords:** Verification complexity · Proof-labeling schemes · CONGEST model · Congested clique

## 1 Introduction

Similarly to classical complexity theory, studying the verification complexity of various problems is one of the major approaches in the quest to understand the complexity of network tasks. The basic idea, proposed by Korman et al. [22] under the name Proof-Labeling Schemes (PLS for short), is to assume that an oracle assigns a label to each node, so that by exchanging these labels, the nodes can collectively verify that a certain global predicate holds (see Sect. 2 for details). The verification complexity of a predicate $\pi$ is defined to be the

P. Spirakis and P. Tsigas (Eds.): SSS 2017, LNCS 10616, pp. 1–17, 2017.
https://doi.org/10.1007/978-3-319-69084-1_1

minimal label length which suffices to verify $\pi$. This node-centric, space-based view was generalized in subsequent work, in which it was allowed for nodes to send different messages to different neighbors, rather than the whole local label to all neighbors. Specifically, in [5] the verification complexity is defined to be the minimal *message*-length required to verify the given predicate.

The distinction between these two models is natural and appears in other contexts as well, like the broadcast and the unicast flavors of congested clique, proposed by Drucker et al. [9]: in the unicast flavor, a node may send a different message to each of its neighbors, while in the broadcast flavor, all neighbors receive the same message. Following up on this model, Becker et al. [6] proposed considering a spectrum of congested clique models, where a node may send up to $r$ distinct messages in a round, where $1 \le r < n$ is a given parameter. This model, called henceforth $\textsc{mcast}(r)$, can be motivated by observing that $r$ can be viewed as the number of network interfaces (NICs) a node possesses: Each interface may be connected to a subset of the neighbors, and it can send only a single message at a time.

**Our Results.** In this paper we present a few preliminary results concerning PLS in the $\textsc{mcast}(r)$ model. Our main focus is on the tradeoff between the number $r$ of different messages a node can send in one round and the verification complexity (message length) $\kappa$. While there are problems whose verification complexity is independent of $r$, we prove that the verification complexity of some fundamental problems is highly dependent on $r$. First, we consider the problem of *matching verification* (MV), where every node has at most one incident edge marked, and the goal is to verify whether the set of marks implies a well defined matching, i.e., an edge is either marked in both endpoints or unmarked in both, and that this set is a matching. In [19], among other results, it is shown that maximal matching has verification complexity $\Theta(1)$, and that the verification complexity of maximum matching in bipartite graphs is also $\Theta(1)$. These results implicitly assume that the subset of edges is well defined; our results show that in fact, the main difficulty is in ensuring that both endpoints of an edge agree on its status. This motivates our next problem that focuses on consistency. Specifically, we define the primitive problem *edge agreement* (EA) as follows. Each node has a $b$-bit string for each incident edge, and a state is considered legal iff both endpoints of each edge agree on the string associated with that edge. It turns out that the *arboricity* of the graph, denoted $\alpha(G)$, plays an important role in the verification complexity of EA (and all problems that EA can locally be reduced to). In Theorem 2, we prove that $\kappa(\textsc{ea}) \cdot r \in \Theta(\alpha(G)b)$. Next, as a more sophisticated example, we consider the important problem of *maximum flow* (MF): In Theorem 3 we show that $\kappa(\textsc{mf}) \cdot r \in \Theta(\alpha(G) \log f_{\max})$, where $f_{\max}$ is the largest flow value over an edge. In [22], a scheme in the broadcast model to verify that the maximum flow between a given pair of nodes $s$ and $t$ is exactly $k$ is given, with complexity $O(k(\log k + \log n))$. We prove, in Theorem 4, that the verification complexity of this problem in the broadcast model is $O(\min\{\alpha(G), k\}(\log k + \log \Delta))$, which is an exponential improvement in some cases. In addition, our upper bound scales linearly with $r$ in the $\textsc{mcast}(r)$ model.

We also consider the congested clique model. To date, no lower bounds on the verification complexity in the congested clique were known. We show that the known technique of crossing [5] can be applied, but only in broadcast clique (i.e., MCAST(1)). We use this argument, along with a new scheme, to obtain a tight $\Theta(\log n + \log w_{\max})$ bound for MST verification in broadcast cliques, where $w_{\max}$ denotes the largest edge weight.

We note that all results translate to randomized PLS [5]. Details of the general connection between the deterministic and randomized verification complexity can be found in the full version [26].

**Related Work.** Drucker et al. [9] propose a *local broadcast* communication in the congested clique, where every node broadcasts a message to all other nodes in each round. Becker et al. [6] proposed, still for congested cliques, a bounded number $r$ of different messages a node can send in each round.

Verification of a given property in decentralized systems finds applications in various domains, such as, checking the result obtained from the execution of a distributed program [4,17], establishing lower bounds on the time required for distributed approximation [8], estimating the complexity of logic required for distributed run-time verification [18], general distributed complexity theory [16], and self stabilizing algorithms [7,21].

The notion of distributed verification in a single round was introduced by Korman et al. in [22]. The verification complexity of minimum spanning-trees (MST) was studied in [20]. Constant-round schemes were studied in [19]. Verification processes in which the global result is not restricted to be the logical conjunction of local outputs had been studied in [2,3]. The role of unique node identifiers in local decision and verification was extensively studied in [13–15]. Proof-labeling schemes in directed networks were studied in [11], where both one-way and two-way communication over directed edges is considered. Verification schemes for dynamic networks, where edges may appear or disappear after label assignment and before verification, are studied in [12]. Recently, a hierarchy of local decision as an interaction between a prover and a disprover was presented in [10].

**Paper Organization.** The remainder of this paper is organized as follows. In Sect. 2 we formalize the model and recall some graph-theoretic concepts. In Sect. 3 we present two general techniques that apply to the MCAST($r$) model. In Sect. 4 we present results for verification of matching, edge agreement, and max-flow. In Sect. 5 we present our results for congested cliques. We conclude in Sect. 6 with some open questions and directions for future work. Many proofs are omitted due to space limitation. They can be found in the full version of the paper [26].

## 2 Model and Preliminaries

**Computational Framework and the MCAST Model.** Our model is derived from the **CONGEST** model [27]. Briefly, a distributed network is modeled as a connected undirected graph $G = (V, E)$, where $V$ is the set of nodes, $E$ is the set of edges, and every node has a unique identifier. In each synchronous round every node performs a local computation, sends a message to each of its neighbors, and receives messages from all neighbors. We denote the number of nodes $|V|$ by $n$ and the number of edges $|E|$ by $m$. For every node $v \in V$, let $d(v)$ be the *degree* of $v$. We denote by $\Delta(G)$ the maximal degree of a node in $G$. We assume that the edges incident to a node $v$ are numbered $1, \ldots, d(v)$.

The main difference between the model considered in this paper, called MCAST$(r)$, and **CONGEST**, is that in MCAST$(r)$ we are given a parameter $r \in \mathbb{N}$ such that a node may send at most $r$ distinct messages simultaneously. More precisely, we assume that prior to sending messages, the neighbors of a node are partitioned into $r$ disjoint subsets (some of which may be empty), such that $v$ sends the same message to all neighbors in a subset. We emphasize that in our model, for simplicity, $r$ is a uniform parameter for all nodes.

**Proof-Labeling Schemes in the MCAST Model.** A *configuration* $G_s$ includes an underlying graph $G = (V, E)$ and a *state* assignment function $s : V \to S$, where $S$ is a (possibly infinite) state space. The state of a node $v$, denoted $s(v)$, includes all local input to $v$. In particular, the state usually includes a unique node identity ID$(v)$ and, in the case of weighted graphs, the weight $w(e)$ of each incident edge $e$. The state of $v$ typically include additional data whose integrity we would like to verify. For example, node state may contain a marking of incident edges, such that the set of marked edges constitutes a spanning tree.

Let $\mathcal{F}$ be a family of configurations, and let $\mathcal{P}$ be a boolean predicate over $\mathcal{F}$. A proof-labeling scheme consists of two conceptual components: a *prover* $\mathbf{p}$, and a *verifier* $\mathbf{v}$. The prover is an oracle which, given any configuration $G_s \in \mathcal{F}$ satisfying $\mathcal{P}$, assigns a bit string $\ell(v)$ to every node $v$, called the *label* of $v$. The verifier is a distributed algorithm running at every node. At each node $v$, the local verifier takes as input the state $s(v)$ of $v$, its label $\ell(v)$ and based on them sends messages to all neighbors. Then, using as input the messages received from the neighbors, the local state and the local label, the local verifier computes a boolean value. If the outputs are TRUE at all nodes, the global verifier $\mathbf{v}$ is said to *accept* the configuration, and otherwise (i.e., at least one local verifier outputs FALSE), $\mathbf{v}$ is said to *reject* the configuration. For correctness, a proof-labeling scheme $\Sigma = (\mathbf{p}, \mathbf{v})$ for $(\mathcal{F}, \mathcal{P})$ must satisfy the following requirements, for every $G_s \in \mathcal{F}$:

- If $\mathcal{P}(G_s) =$ TRUE then, using the labels assigned by $\mathbf{p}$, the verifier $\mathbf{v}$ accepts $G_s$.
- If $\mathcal{P}(G_s) =$ FALSE then, for every label assignment, the verifier $\mathbf{v}$ rejects $G_s$.

Given a configuration $G_s$, we denote by $\boldsymbol{c}_{\Sigma}(G_s)$ the vector of length $|E|$ that contains the messages sent according to the scheme $\Sigma$, and we refer to this vector as the *communication pattern* of $\Sigma$ over $G_s$. For an underlying graph $G$, we denote by $L(G)$ the number of legal configurations of $G$, and by $W_\Sigma(G)$ the number of different communication patterns of $\Sigma$ in $G$, over all legal configurations. In our analysis, given an edge $(v, u) \in E$, we denote by $M_v(e)$ the message over $e$ from $v$ to $u$.

Our central measure for PLSs is its verification complexity, defined as follows.

**Definition 1.** *The* verification complexity *of a proof labeling scheme $\Sigma = (\mathbf{p}, \mathbf{v})$ for the predicate $\mathcal{P}$ over a family of configurations $\mathcal{F}$ is the maximal length of a message generated by the verifier $\mathbf{v}$ based on the labels assigned to the nodes by the prover $\mathbf{p}$ in a configuration $G_s$ for which $\mathcal{P}(G_s) = \text{TRUE}$.*

In this paper we consider PLSs in the MCAST$(r)$ model, namely we impose the additional restriction that at most $r$ distinct messages may be sent by a node.

**Arboricity, Degeneracy and Average Degree.** The average degree of a graph plays a central role in our study. However, graphs may have dense and sparse regions. We therefore use the following refined concepts.

**Definition 2.** *The* arboricity *of a graph $G = (V, E)$, denoted by $\alpha(G)$, is defined as the minimum number of acyclic subsets of edges that cover $E$. The* degeneracy *of a graph $G$, denoted by $\delta(G)$, is defined as the smallest value $i$ such that the edges of $G$ can be oriented to form a directed acyclic graph with out-degree at most $i$.*

The following properties are well known [24,25].

**Lemma 1.** *For all graphs $G$, $\alpha(G) \leq \delta(G) < 2\alpha(G)$.*

**Lemma 2.** *For a given graph $G = (V, E)$, $\alpha(G) = \max\left\{ \left\lceil \frac{m_H}{n_H - 1} \right\rceil \mid V_H \subseteq V, |V_H| \geq 2 \right\}$, where $m_H = |E_H|$ and $n_H = |V_H|$ over all induced subgraphs $H = (V_H, E_H)$ of $G$.*[1]

Note that by Lemmas 1 and 2, the minimal number of outgoing edges in the best orientation of a graph $G$ is proportional to the maximal average degree over all induced subgraphs of $G$.

## 3 Techniques for the MCAST Model

In this work, we consider problems expressible as a conjunction of edge predicates, where a node may have a different input for every edge. We present two techniques that can be used as building blocks in the design of efficient PLSs in the MCAST model.

[1] Given a graph $G = (V, E)$, the *induced subgraph* $H = (V_H, E_H)$ over the set of nodes $V_H \subseteq V$ satisfies that $E_H = E \cap (V_H \times V_H)$.

The first technique, which we call *minimizing orientation*, reduces the number of incident edges a node sends its input on. We orient the edges such that the maximum out degree is minimized. Lemma 1 ensures that the maximum out degree is bounded by $2\alpha$. Using a minimizing orientation, we can prove the following lemma.

**Lemma 3.** *Suppose that a verification problem* $(\mathcal{F}, \mathcal{P})$ *is expressible as a conjunction of edge predicates, each involving variables from a single pair of neighbors. Then there exists a PLS* $\Sigma = (\mathbf{p}, \mathbf{v})$ *for* $(\mathcal{F}, \mathcal{P})$ *in the* $\textsc{mcast}(2\alpha)$ *model with verification complexity* $k$*, where* $k$ *is the length of the largest local input to an edge predicate.*

*Color Addressing.* In the unicast model, each node receives its own message. However, if we want to use a unicast PLS in the $\textsc{mcast}(r)$ model with $r < 2\alpha$, we may need to bundle together a few messages, and hence we need to somehow tag each part of the message with its intended recipient. Clearly this can be done by tagging each sub-message by the unique ID of recipient, but this adds $\Theta(\log n)$ bits to each sub-message. The *color addressing* technique reduces this overhead to $O(\log \Delta)$. The idea is that each node need only distinguish between its neighbors.[2] We solve this difficulty by coloring the nodes so that no two neighbors of a node get the same color. Formally, *color addressing* is a PLS $\Sigma_{COL} = (\mathbf{p}, \mathbf{v})$ in the broadcast model, where the prover $\mathbf{p}$ first colors the nodes so that no two nodes at distance 1 or 2 receive the same color. This is possible using at most $\Delta^2 + \Delta + 1 \in O(\Delta^2)$ colors, because every node has at most $\Delta$ neighbors and $\Delta^2$ nodes at distance 2 from it. Next, the prover assigns to every incident edge of a node the color of the neighbor at the other end of the edge. The verifier $\mathbf{v}$ at a node $v$ broadcasts the color assigned to $v$ by the prover. Every node verifies that every incident edge is assigned a different color and that the color received from every edge is the color assigned by the prover to this edge.

Clearly, $\Sigma_{COL}$ guarantees a proper coloring as desired to use for addressing, and this coloring is locally verifiable. Moreover, since a color can be represented using $O(\log \Delta)$ bits, we obtain local addressing with verification complexity $O(\log \Delta)$ in the *broadcast* model. We summarize in the following lemma.

**Lemma 4.** $\Sigma_{COL}$ *is a PLS in the broadcast model, which assigns and verifies an* $O(\log \Delta)$*-bit coloring for proper addressing. The verification complexity of* $\Sigma_{COL}$ *is* $O(\log \Delta)$*.*

## 4 Verification Complexity Trade-Offs in the $\textsc{mcast}(r)$ Model

In this section, we study the effect of $r$ on the verification complexity of PLSs in the $\textsc{mcast}(r)$ model. We start with the observation that for some problems, the

[2] We note that using simple port numbering requires agreement with the neighbors, which is costly, as we prove in Theorem 2.

asymptotic verification complexity is independent of $r$. These problems include the deterministic verification of a spanning-tree and vertex bi-connectivity, and the randomized verification of an MST. For each of these problems, we provide a scheme for $r = 1$ with verification complexity that matches the lower bound for $r = \Delta$ [5,22]. In contrast, there are problems for which the verification complexity is sensitive to $r$. Specifically, we present a tight bound for the matching verification problem in the broadcast model, which is reduced dramatically even for $r = 2$. Finally, we show tight bounds for the primitive problem of edge agreement and the more sophisticated application of maximum flow, which scales linearly with $r$.

### 4.1 Verification of Matchings

In the literature, in verification problems of the form "does a subset of edges satisfy a specified property," it is usually assumed that the subset of edges is well defined, i.e., for every edge $e = (u, v)$, the local state of $v$ indicates that $e$ is in the subset if and only if the local state of $u$ indicates it. However, since edges do not have storage, an edge set is actually represented by the local state at the nodes, and hence consistency between neighbors is not always guaranteed.

In fact, there are problems for which the verification of consistency is the dominant factor of the verification complexity. In particular, consider matching problems: maximal matching, and maximum matching in bipartite graphs. Both problems are known to have constant verification complexity [19]. However, these results make the problematic assumption that the edge set in question is well defined. We consider the matching verification problem using the following definition.

**Definition 3 (Matching Verification (MV)).**
***Instance****: At each node $v$, at most one edge is marked. We use $I_v(e) \in \{\text{TRUE}, \text{FALSE}\}$ to denote whether $e$ is marked in $v$.*
***Question****: Is the set $M$ of marked edges well defined, i.e., $I_v(e) = I_u(e)$ for every edge $e = (u, v) \in E$, and $M$ is a matching?*

We argue that in the broadcast model, the verification complexity of this problem is $\Theta(\log \Delta)$. Formally, we study the problem $(\mathcal{F}_m, \text{MV})$, where $\mathcal{F}_m$ is the family of connected configurations with edge indication at each node. We obtain the following result.

**Theorem 1.** *The verification complexity of $(\mathcal{F}_m, \text{MV})$ in the broadcast model is $\Theta(\log \Delta)$.*

For the lower bound, we construct a set of configurations that must have different communication patterns. The large number of configurations implies the lower bound on message length. We use color addressing for the upper bound.

The result above says that in the broadcast model, the verification complexity of the maximal matching problem and the maximum matching in bipartite graphs is dominated by the consistency verification. Observe that in the

MCAST(2) model, the verification complexity of $(\mathcal{F}_m, \textsc{mv})$ is $O(1)$, by letting every node $v$ send on every edge $e = (v, u)$ the bit $I_v(e)$: only two types of messages are needed!

We also note that for the problem of maximum matching in cycles, the asymptotic verification complexity is unchanged if we must verify consistency, since the verification complexity of this problem in the broadcast model is $\Theta(\log n)$ [19].

### 4.2 The Edge Agreement Problem

Motivated by the results for matching verification, we now formalize and study the fundamental problem of consistency across edges.

**Definition 4. ($b$-bit Edge Agreement ($\textsc{ea}_\mathbf{b}$)).**
***Instance:*** *Each node $v$ holds in its state a $b$-bit string $B_v(e)$ for each incident edge $e$.*
***Question:*** *Is $B_v(e) = B_u(e)$ for every edge $e = (u, v) \in E$?*

Let $\mathcal{F}$ be the family of all configurations, and let $\alpha$ denote the arboricity of the graph. Our first main result is the following tight trade-off between $r$ (the number of different messages for a node) and verification complexity of $\textsc{ea}_b$.

**Theorem 2.** *Let $b \in \Omega(\log \Delta)$. For every $1 \le r \le \min\left\{\Delta, 2^{b/4}\right\}$, the verification complexity of $(\mathcal{F}, \textsc{ea}_b)$ in the* MCAST$(r)$ *model is $\Theta(\left\lceil \frac{\alpha}{r} \right\rceil b)$.*

This theorem states both an upper and a lower bound. We start with the lower bound.

**Lemma 5.** *For every $1 \le r \le \min\left\{\Delta, 2^{b/4}\right\}$, the verification complexity of any PLS for $(\mathcal{F}, \textsc{ea}_b)$ in the* MCAST$(r)$ *model is $\Omega((\frac{\alpha}{r} + 1)b)$.*

To prove Lemma 5, we prove the following claim.

*Claim.* Let $G = (V, E)$ be a graph, let $1 \le r \le \min\left\{\Delta, 2^{b/4}\right\}$ and consider a PLS for $(\mathcal{F}, \textsc{ea}_b)$ in the MCAST$(r)$ model. For every induced subgraph $H = (V_H, E_H)$ of $G$, $W_\Sigma(H) \ge L(H)$.

**Proof of Lemma 5:** It is known that the non-deterministic two-party communication complexity of verifying the equality (EQ) of $b$-bit strings is $\Omega(b)$ [23, Example 2.5]. Simulating a verification scheme for $(\mathcal{F}, \textsc{ea}_b)$ on a network of one edge, is a correct non-deterministic two-party communication protocol for EQ. Therefore, $\Omega(b)$ is a lower bound for $(\mathcal{F}, \textsc{ea}_b)$.

We now prove that $\Omega(\frac{\alpha}{r} b)$ is also a lower bound for $(\mathcal{F}, \textsc{ea}_b)$. Let $G_s \in \mathcal{F}$ be a configuration with an underlying graph $G = (V, E)$, and let $H = (V_H, E_H)$ be the densest induced subgraph of $G$, i.e., $m_H/n_H \ge m_{H'}/n_{H'}$ for every $V'_H \subseteq V$. By Lemma 2, $\alpha = \lceil m_H/(n_H - 1) \rceil$. W.l.o.g., let $V_H = \{v_1, \dots, v_{n_H}\}$, and let $d_H(v_i) = |\{(v_i, v_j) \in E_H\}|$ be the degree of node $v_i$ in $H$.

We now show that for $1 \leq r \leq \min\left\{\Delta, 2^{b/4}\right\}$ and any scheme $\Sigma$ for $(\mathcal{F}, \text{EA}_b)$ with verification complexity $\kappa < \frac{\alpha b}{4r} - 2$ in the MCAST$(r)$ model, it holds that $W_\Sigma(H) < L(H)$. Let $\Sigma$ be such a verification scheme. Then

$$\begin{aligned} W_\Sigma(H) &\leq \prod_{i=1}^{n_H}\left[\binom{2^\kappa}{r} \cdot r^{d_H(v_i)}\right] && (1)\\ &\leq \left(\frac{2^\kappa \cdot e}{r}\right)^{rn_H} \cdot r^{2m_H} && (2)\\ &< 2^{\alpha b n_H/4} \cdot r^{2m_H} && (3)\\ &\leq 2^{\frac{b}{2}m_H} \cdot r^{2m_H} && (4)\\ &\leq 2^{bm_H} = L(H). && (5)\end{aligned}$$

Inequality (1) is true since for every PLS in the MCAST$(r)$ model with verification complexity $\kappa$, every communication pattern can be constructed by letting each node $v_i$ choose $r$ different messages of size $\kappa$ each, and for each of its $d_H(v_i)$ neighbors, let it choose one of the $r$ messages to send. Inequality (2) is due to the fact that $\binom{x}{y} \leq (\frac{x \cdot e}{y})^y$ for $x, y \geq 0$. Inequality (3) follows from our assumption that $\kappa < \frac{\alpha b}{4r} - 2$. Inequality (4) follows from Lemma 2 which implies that $\alpha \leq 2m_H/n_H$, and Inequality (5) from our assumption that $r \leq 2^{b/4}$.

Therefore we may conclude that if $\kappa < \frac{\alpha b}{4r} - 2$, then, by Claim 4.2, $\Sigma$ is not a correct verification scheme for $(\mathcal{F}, \text{EA}_b)$. This concludes the proof of the lower bound. ∎

Next, we turn to the upper bound. To this end we define a more general problem as follows.

**Definition 5 ($b$-bit Edge $\psi$ (E$\psi_\mathbf{b}$)).**
***Instance:*** *Each node $v$ holds in its state a $b$-bit string $B_v(e)$ for each incident edge $e$.*
***Question:*** *Is $\psi_b(B_v(e), B_u(e)) =$ TRUE for every edge $e = (u, v)$, where $\psi_b$ is a given symmetric predicate of two $b$-bit strings, i.e., $\psi_b : \{0,1\}^b \times \{0,1\}^b \to \{\text{TRUE}, \text{FALSE}\}$ and $\psi(s, s') = \psi(s', s)$ for all $s, s' \in \{0,1\}^b$?*

**Lemma 6.** *For every $1 \leq r < 2\alpha$, there exists a PLS for $(\mathcal{F}, \text{E}\psi_b)$ in the MCAST$(r)$ model with verification complexity $O(\frac{\alpha}{r}(b + \log\Delta))$, and for every $2\alpha \leq r \leq \Delta$, there exists a PLS for $(\mathcal{F}, \text{E}\psi_b)$ in the MCAST$(r)$ model with verification complexity $O(b)$.*

We sketch the proof of Lemma 6. For $1 \leq r < 2\alpha$, we use minimizing orientation and color addressing. The idea is to partition the outgoing edges into $r$ groups, and send the input strings of every group in one message, indicating the color of the destination of each string. Overall, every message consists of at most $2\alpha/r$ pairs of size $b + O(\log\Delta)$ each. For $2\alpha \leq r \leq \Delta$, by Lemma 3 there exists a PLS $\Sigma' = (\mathbf{p}', \mathbf{v}')$ for $(\mathcal{F}, \text{E}\psi_b)$ in the MCAST$(r)$ model with verification complexity $b$.

$\text{EA}_b$ is a special case of $\text{E}\psi_b$, where $\psi$ is the equality predicate. Therefore, Lemma 6 gives a tight upper bound for $(\mathcal{F}, \text{EA}_b)$ for the case $b \in \Omega(\log \Delta)$. This concludes the proof of Theorem 2.

We note that Theorem 2, in conjunction with the general connection between the deterministic and randomized verification complexity [26], gives the following corollary.

**Corollary 1.** *Let $b \in \Omega(\log \Delta)$. For every $1 \le r \le \min\left\{\Delta, 2^{b/4}\right\}$, the randomized verification complexity of $(\mathcal{F}, \text{EA}_b)$ in the* MCAST$(r)$ *model is $\Theta(\log(\lceil \frac{\alpha}{r} \rceil b))$.*

### 4.3 An Advanced Example: The Maximum Flow Problem

In this section we consider a more sophisticated problem, namely Maximum Flow in the context of the MCAST$(r)$ model. The best previously known result [22] was for verification of "$k$-flow": the goal is to verify that the maximum flow between a given pair of nodes is exactly $k$. The verification complexity of the scheme in the broadcast model of [22] is $O(k(\log k + \log n))$. In Theorem 4, we show an improvement of this result and a generalization to the MCAST$(r)$ model.

First, we solve a slightly different problem, formalized as follows. Let $\mathcal{F}_{st}$ be the family of configurations of graphs, where a graph in $\mathcal{F}_{st}$ has two distinct nodes denoted $s$ and $t$ called *source* and *sink*, respectively, and a natural number $c(e)$ called the *capacity* associated with each edge $e$. The MF problem is defined over the family of configurations $\mathcal{F}_{st}$ as follows.

**Definition 6 (Maximum Flow (MF)).**
***Instance:*** *A configuration $G_s \in \mathcal{F}_{st}$, where each node $v$ has an integer $f(v,u)$ for every neighbor $u$.*
***Question:*** *Interpreting $f(v,u)$ as the amount of flow from $v$ to $u$ ($f(v,u) < 0$ means flow from $u$ to $v$), is $f$ a maximum flow from $s$ to $t$?*

Recall that $f$ is a legal flow iff it satisfies the following three conditions (see, e.g., [1]).

- Anti symmetry: for every $(v,u) \in E$, $f(v,u) = -f(u,v)$.
- Capacity compliance: for every $(v,u) \in E$, $|f(v,u)| \le c(v,u)$.
- Flow conservation: for every node $v \in V \setminus \{s,t\}$, $\sum_{u \in V} f(v,u) = 0$.

If all three conditions hold, then, by the max-flow min-cut theorem, $f$ is maximum iff there is a saturated cut.

We denote by $f_{\max}$ the maximal flow amount over all edges of $G$ (note that $f_{\max}$ need not be polynomial in $n$). Also, for a bit string $x = x_0 x_1 \cdots x_k$, let $\bar{x} = \sum_{i=0}^{k} x_i 2^i$.

**Theorem 3.** *Let $\log f_{\max} \in \Omega(\log n)$. There exists a constant $c > 1$ such that for every $1 \le r \le \min\left\{\alpha/c, \sqrt[4]{f_{\max}}\right\}$, the verification complexity of $(\mathcal{F}_{st}, \text{MF})$ in the* MCAST$(r)$ *model is $\Theta(\log(f_{\max})\alpha/r)$.*

Again, we start with the lower bound.

**Lemma 7.** *Let $\log f_{\max} \in \Omega(\log n)$. There exists a constant $c > 1$ such that for every $1 \leq r \leq \min\left\{\alpha/c, \sqrt[4]{f_{\max}}\right\}$, the verification complexity of any PLS for $(\mathcal{F}_{st}, \textsc{mf})$ in the* $\textsc{mcast}(r)$ *model is $\Omega(\log(f_{\max})\alpha/r)$.*

We note that the counting argument used for $\textsc{ea}_b$ (Lemma 5) cannot be applied to this problem. To prove the lower bound for $\textsc{mf}$, we show a non-trivial reduction from a problem in $(\mathcal{F}, \textsc{ea}_b)$ to a problem in $(\mathcal{F}_{st}, \textsc{mf})$.

**Lemma 8.** *For every $1 \leq r < 2\alpha$, there exists a PLS for $(\mathcal{F}_{st}, \textsc{mf})$ in the* $\textsc{mcast}(r)$ *model with verification complexity $O(\frac{\alpha}{r}(\log f_{\max} + \log \Delta))$, and for every $2\alpha \leq r \leq \Delta$, there exists a PLS for $(\mathcal{F}_{st}, \textsc{mf})$ in the* $\textsc{mcast}(r)$ *model with verification complexity $O(\log f_{\max})$.*

The scheme used in the proof of Lemma 8 consists of two parts. First, a scheme for $\psi$ agreement, where $\psi(x, y) \equiv (\overline{x} = -\overline{y})$, which, we argue, is enough in order to verify that the flow is legal. The second part is verifying a saturated $s$-$t$ cut. This can be done using one bit at each node.

For $\log f_{\max} \in \Omega(\log n)$, Lemma 8 gives a tight upper bound for $(\mathcal{F}_{st}, \textsc{mf})$ which concludes the proof of Theorem 3.

Consider now the $k$-$\textsc{mf}$ problem as defined in [22] over the family of configurations $\mathcal{F}_{st}$.

**Definition 7 ($k$-Maximum Flow ($k$-MF)).**
***Instance:*** *A configuration $G_s \in \mathcal{F}_{st}$.*
***Question:*** *Is the maximum flow between $s$ and $t$ in $G_s$ is exactly $k$?*

We give an upper bound for $(\mathcal{F}_{st}, k\text{-}\textsc{mf})$ in the $\textsc{mcast}(r)$ model, which generalizes and improves the previous bound.

**Theorem 4.** *For every $1 \leq r < 2\alpha$, there exists a PLS for $(\mathcal{F}_{st}, k\text{-}\textsc{mf})$ in the* $\textsc{mcast}(r)$ *model, with verification complexity $O\left(\frac{\min\{\alpha,k\}}{r}(\log k + \log \Delta)\right)$, and for every $2\alpha \leq r \leq \Delta$, there exists a PLS for $(\mathcal{F}_{st}, k\text{-}\textsc{mf})$ in the* $\textsc{mcast}(r)$ *model, with verification complexity $O(\log k)$.*

**Proof:** In a verification scheme for $(\mathcal{F}_{st}, k\text{-}\textsc{mf})$, the prover can assign the flow values $f(v, u)$ for every edge $(v, u)$. W.l.o.g, assume that $f$ does not contain cycles of positive flow. In this case, $f_{\max} \leq k$ and, since the flow value over each edge is an integer, the number of incident edges of every node $v$ carrying non-zero flow is at most $2k$. By Lemma 8, and the observation that it is sufficient that every node verifies the value of flow only on edges with $f(v, u) \neq 0$, the upper bounds follow. ∎

To be precise, the problem solved in [22] required in addition that every node holds the value $k$ in its state. Verifying that all nodes hold the same value $k$ is simply an additive $\log k$ factor to message length – every node sends its value and verifies that all its neighbors have the same value. We argue in the following lemma, that $\Omega(\log k)$ is a lower bound for $(\mathcal{F}_{st}, k\text{-}\textsc{mf})$ verification even if $k$ is known to all nodes.

**Lemma 9.** *For every $1 \leq k \leq 2^{\Theta(n)}$, the verification complexity of any PLS for $(\mathcal{F}_{st}, k\text{-MF})$ is $\Omega(\log k)$, even in the unicast model and for constant degree graphs.*

We use a kind of crossing argument between a family of different configurations of the same structure, to show that a scheme with verification complexity less than $\frac{\log k}{4}$ is never a correct scheme for all configurations in the constructed family. Hence, the lower bound follows.

By Theorem 4, this lower bound is tight for $2\alpha \leq r \leq \Delta$, and the following theorem holds.

**Theorem 5.** *For every $1 \leq k \leq 2^{\Theta(n)}$ and every $2\alpha \leq r \leq \Delta$, the verification complexity of $(\mathcal{F}_{st}, k\text{-MF})$ in the* MCAST$(r)$ *model is $\Theta(\log k)$.*

# 5 Verification in Congested Cliques

In the congested clique model, the communication network is a fully connected graph over $n$ nodes (i.e., an $n$-clique). Given an input graph $G = (V, E)$ with $n = |V|$, the nodes of $G$ are mapped 1–1 to the nodes of the clique, and the state of each node contains a bit for each port, indicating whether the edge to that port is in $E$ or not, and, if the edge is present and $G$ is weighted, the weight of the edge. We assume that the part in the state that specifies whether the edge connected to this port is in $E$ is reliable: since verification is done with respect to the given graph as input, there is no way to verify its authenticity, but only whether the combination of input and output satisfies the given predicate. Moreover, we assume that the input is consistent, in the sense that the state at node $v$ indicates that $(v, u)$ is an edge in $E$ (possibly with some weight $w$), if and only if so does the state of $u$ (namely edge agreement on the input graph is guaranteed).

## 5.1 Crossing in Congested Cliques

In what follows, we say that an edge is *oriented* to indicate a specific order over its endpoints.

**Definition 8 (Independent Edges).** *Let $G = (V, E)$ be a graph and let $e_1 = (v_1, u_1)$ and $e_2 = (v_2, u_2)$ be two oriented edges of $G$. The edges $e_1$ and $e_2$ are said to be* independent *if and only if $v_1, u_1, v_2, u_2$ are four distinct nodes and $(v_1, u_2), (v_2, u_1) \notin E$.*

The following definition is illustrated in Fig. 1.

**Definition 9 (Crossing [5]).** *Let $G = (V, E)$ be a graph, let $e_1 = (v_1, u_1)$ and $e_2 = (v_2, u_2)$ be two independent oriented edges of $G$, and for $i \in \{1, 2\}$, let $p_i$ and $q_i$ be the port numbers of $e_i$ at $v_i$ and $u_i$ respectively. The* crossing *of $e_1$ and $e_2$ in $G$, denoted by $G(e_1, e_2)$, is the graph obtained from $G$ by replacing $e_1$ and $e_2$ with the edges $e'_1 = (v_1, u_2)$ and $e'_2 = (v_2, u_1)$ so that $e'_1$ connects port $p_1$ at $v_1$ and port $q_2$ at $u_2$ and $e'_2$ connects port $p_2$ at $v_2$ and port $q_1$ at $u_1$.*

Consider an input graph $G = (V, E)$ in the clique, assume that $e_1, e_2 \in E$ are independent edges and let $G(e_1, e_2) = (V, E')$. Note that crossing a graph over a clique network does not result in a change of state: Due to the port preservation of the crossing operation, for every node $v \in V$ and every port $0 \leq i \leq n-1$, the edge $(v, u)$ on port number $i$ in $G$ satisfies $(v, u) \in E$ if and only if the edge $(v, u')$ on port number $i$ in $G(e_1, e_2)$ satisfies $(v, u') \in E'$.

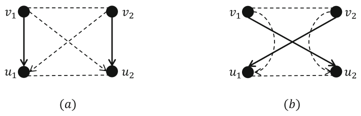

**Fig. 1.** An illustration of the crossing operation on a clique network. Solid edges are input graph edges, and dashed edged are communication-only edges. (a) Edges $e_1 = (v_1, u_1)$ and $e_2 = (v_2, u_2)$ are two independent oriented edges of an input graph $G$. (b) The subgraph induced by nodes $v_1, u_1, v_2$ and $u_2$ in $G(e_1, e_2)$.

Whether we can prove a lower bound for verification in the congested clique for $r > 1$ is still an open question. However, for the broadcast clique model (i.e., $r = 1$), it turns out that we can. The following lemma is the key to proving lower bounds for PLSs in the broadcast clique.

**Lemma 10.** *Let $\mathcal{F}$ be a family of configurations, let $\mathcal{P}$ be a boolean predicate over $\mathcal{F}$, and let $\Sigma$ be a PLS for $(\mathcal{F}, \mathcal{P})$ in the broadcast clique model with verification complexity $\kappa$. Suppose that there is a configuration $G_s \in \mathcal{F}$ such that $\mathcal{P}(G_s) = \textsc{true}$ and $G$ contains $q$ pairwise independent oriented edges $e_1, \ldots, e_q$. If $\kappa < \frac{\log q}{2}$, then there are $1 \leq i < j \leq q$ such that $G_s(e_i, e_j)$ is accepted by $\Sigma$.*

In the proof of this lemma, we show that in the broadcast clique, if verification complexity is too small, then we can apply the pigeonhole principle on the crossing of every two edges from the set. We get that there must be two edges such that the local view of all nodes is the same for the original input graph and the crossed graph. Therefore, we conclude that with the same label assignment, both configurations (original and crossed) result in the same output.

We use the following corollary of Lemma 10 to lower-bound verification complexity of broadcast clique PLSs.

**Corollary 2.** *Let $\mathcal{F}$ be a family of configurations, and let $\mathcal{P}$ be a boolean predicate over $\mathcal{F}$. If there is a configuration $G_s \in \mathcal{F}$ satisfying that $\mathcal{P}(G_s) = \textsc{true}$ and $G$ contains $q$ pairwise independent oriented edges $e_1, \ldots, e_q$ such that for every $1 \leq i < j \leq q$ it holds that $\mathcal{P}(G_s(e_i, e_j)) = \textsc{false}$, then the verification complexity of any deterministic PLS for $(\mathcal{F}, \mathcal{P})$ in the broadcast clique model is $\Omega(\log q)$.*

Note that we essentially cross two pairs of edges in the crossing operation: one pair of edges in $E$, and one pair of edges in $\bar{E}$. These two pairs are uniquely associated with each other in a way that if we assume a PLS in the $\textsc{mcast}(2)$ clique model, then we would not be able to apply the pigeonhole principle even with 1-bit messages. To see why this is true, consider any set of independent oriented edges $(v_1, u_1), \ldots, (v_q, u_q)$. For every $i \neq j$, both edges $(v_i, u_j), (v_j, u_i) \in \bar{E}$ are associated only with the pair of edges $(v_i, u_i), (v_j, u_j) \in E$. Therefore, with a PLS in the MCAST(2) clique model, it is possible that $M_{v_i}(u_j) \neq M_{v_j}(u_i)$ for every $i \neq j$ independently of other pairs. Hence, the crossing of any two edges may change the local view of at least one node. Therefore, the crossing technique can not be applied for every $r > 1$ in the congested clique.

### 5.2 Minimum Spanning-Tree Verification

In this section we illustrate the use of Corollary 2 and prove tight bounds for the verification complexity of the Minimum Spanning-Tree (MST) problem. Recall that an MST of a weighted graph $G$ is a spanning tree of $G$ whose sum of all its edge-weights is minimum among all spanning trees of $G$. In particular, in the clique, there is a fully connected communication network, a weighted input graph $G = (V, E, w)$ where $E$ is a subset of communication edges, $w : E \to \mathbb{N}$ is the edge weight assignment, and a subset $T \subseteq E$ is specified as the MST. It is important to notice that all specifications of edge subsets are local in the sense that every node $v \in V$ has $n-1$ ports and in its state there is a specification for every edge $e_i$ on port number $i$ whether $e_i \in E$ and whether $e_i \in T$. According to our assumption on the clique model, the input graph $G$ is given in a reliable way, i.e., an edge $(v, u)$ is considered by $v$ to be in $E$ if and only if it is considered by $u$ to be in $E$. However, this consistency has to be verified for the edges of $T$. In addition, since the communication network is fully connected and does not depend on the input graph $G$, we also consider the case where $G$ is disconnected. In this case, we define the MST as the set of minimum spanning-trees of all connected components of $G$.

Let $\mathcal{F}_{w_{\max}}$ be the family of all weighted configurations (not necessarily connected) with maximum weight $w_{\max}$. Formally, if $e$ is an edge of the underlying weighted graph of a configuration $G_s \in \mathcal{F}_{w_{\max}}$, then $w(e) \leq w_{\max}$. Edge weights are assumed to be known at their endpoints.

**Theorem 6.** *The verification complexity of $(\mathcal{F}_{w_{\max}}, MST)$ in the broadcast clique model is $\Theta(\log n + \log w_{\max})$.*

The lower bound is proved in two parts. To show $\Omega(\log n)$ we use Corollary 2 on the input graph which is a path where all the edges are in $T$. The crossing of every two independent edges of the path results in a graph with a cycle component, in particular, not a tree. The $\Omega(\log w_{\max})$ part is proven by a variation of the $\Omega(\log w_{\max})$ proof in [22], which holds also for the broadcast clique model. The tight upper bound is obtained by a scheme for which we give a short sketch here. The prover roots the tree and give every node a pointer to its parent. For verification, every node sends the information about the edge connecting it to its parent – IDs of the endpoints and the weight of the edge. This enables every node $v$ to collect all the tree structure, and verify that if an incident edge $(v, u)$ is not in the tree then its weight is not smaller than every edge in the unique path between $v$ and $u$ in the tree. If all nodes verify this property, it means that all edges are consistent with the "red rule", i.e., the heaviest edge of every cycle is not in the MST.

## 6 Conclusion

In this paper we studied the MCAST($r$) model from the perspective of verification. This angle seems particularly convenient, because it involves a single round of message exchange. (If multiple rounds are allowed, one has to consider the possibility of reconfiguring the neighbor partitions: is it allowed to partition the neighbors anew in each round, and if so, at what cost?). We focus on the relation between the number of different messages of each node and the verification complexity of proof-labeling schemes. We gave tight bounds on the verification complexity of edge agreement and max flow in the MCAST($r$) model. We have shown that in the restrictive broadcast model, a well defined matching is harder to verify than the maximality of a given matching, and that it is possible to obtain lower bounds on the verification complexity in congested cliques. Many interesting questions remain open. We list a few below.

- Develop a theory for a restricted number of interface cards (NICs). The number of NICs limits the number of messages that can be simultaneously transmitted. In this paper we looked only at a simple case of one round of communication. We believe that developing a tractable and realistic model in which the number of NICs is a parameter is an important challenge.
- As mentioned, in multiple round algorithms, dynamic reconfigurations can be exploited to convey information. It seems that an interesting challenge would be to account for dynamic reconfigurations.
- We considered a model in which a single parameter $r$ is used to indicate the restriction of all nodes. What can be said about a model in which every node has its own restriction?
- We have given examples of problems that have a linear improvement in verification complexity as a function of $r$, and on the other hand, we have given examples of problems that are not sensitive at all to $r$. Can a characterization of problems be shown, according to their sensitivity of verification complexity to $r$?

## References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows. Prentice-Hall, Engelwood Cliffs (1993)
2. Arfaoui, H., Fraigniaud, P., Ilcinkas, D., Mathieu, F.: Distributedly testing cycle-freeness. In: Kratsch, D., Todinca, I. (eds.) WG 2014. LNCS, vol. 8747, pp. 15–28. Springer, Cham (2014). doi:10.1007/978-3-319-12340-0_2
3. Arfaoui, H., Fraigniaud, P., Pelc, A.: Local decision and verification with bounded-size outputs. In: Higashino, T., Katayama, Y., Masuzawa, T., Potop-Butucaru, M., Yamashita, M. (eds.) SSS 2013. LNCS, vol. 8255, pp. 133–147. Springer, Cham (2013). doi:10.1007/978-3-319-03089-0_10
4. Awerbuch, B., Patt-Shamir, B., Varghese, G.: Self-stabilization by local checking and correction. In: 32nd Symposium on Foundations of Computer Science (FOCS), pp. 268–277. IEEE (1991)

5. Baruch, M., Fraigniaud, P., Patt-Shamir, B.: Randomized proof-labeling schemes. In: Proceedings of 34th ACM Symposium on Principles of Distributed Computing (PODC), pp. 315–324 (2015)
6. Becker, F., Anta, A.F., Rapaport, I., Rémila, E.: The effect of range and bandwidth on the round complexity in the congested clique model. In: Dinh, T.N., Thai, M.T. (eds.) COCOON 2016. LNCS, vol. 9797, pp. 182–193. Springer, Cham (2016). doi:10.1007/978-3-319-42634-1_15
7. Blin, L., Fraigniaud, P., Patt-Shamir, B.: On proof-labeling schemes versus silent self-stabilizing algorithms. In: Felber, P., Garg, V. (eds.) SSS 2014. LNCS, vol. 8756, pp. 18–32. Springer, Cham (2014). doi:10.1007/978-3-319-11764-5_2
8. Das Sarma, A., Holzer, S., Kor, L., Korman, A., Nanongkai, D., Pandurangan, G., Peleg, D., Wattenhofer, R.: Distributed verification and hardness of distributed approximation. SIAM J. Comput. **41**(5), 1235–1265 (2012)
9. Drucker, A., Kuhn, F., Oshman, R.: On the power of the congested clique model. In: Proceedings of 2014 ACM Symposium on Principles of Distributed Computing, PODC 2014, pp. 367–376. ACM, New York (2014)
10. Feuilloley, L., Fraigniaud, P., Hirvonen, J.: A hierarchy of local decision. In: 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016), pp. 118:1–118:15. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)
11. Foerster, K.-T., Luedi, T., Seidel, J., Wattenhofer, R.: Local checkability, no strings attached. In: Proceedings of 17th International Conference on Distributed Computing and Networking, ICDCN 2016, pp. 21:1–21:10. ACM, New York (2016)
12. Foerster, K.-T., Richter, O., Seidel, J., Wattenhofer, R.: Local checkability in dynamic networks. In: Proceedings of 18th International Conference on Distributed Computing and Networking, ICDCN 2017, pp. 4:1–4:10. ACM, New York (2017)
13. Fraigniaud, P., Göös, M., Korman, A., Suomela, J.: What can be decided locally without identifiers? In: Proceedings of 2013 ACM Symposium on Principles of Distributed Computing, PODC 2013, pp. 157–165. ACM, New York (2013)
14. Fraigniaud, P., Halldórsson, M.M., Korman, A.: On the impact of identifiers on local decision. In: Baldoni, R., Flocchini, P., Binoy, R. (eds.) OPODIS 2012. LNCS, vol. 7702, pp. 224–238. Springer, Heidelberg (2012). doi:10.1007/978-3-642-35476-2_16
15. Fraigniaud, P., Hirvonen, J., Suomela, J.: Node labels in local decision. In: Scheideler, C. (ed.) Structural Information and Communication Complexity. LNCS, vol. 9439, pp. 31–45. Springer, Cham (2015). doi:10.1007/978-3-319-25258-2_3
16. Fraigniaud, P., Korman, A., Peleg, D.: Towards a complexity theory for local distributed computing. J. ACM **60**(5), 35 (2013)
17. Fraigniaud, P., Rajsbaum, S., Travers, C.: Locality and checkability in wait-free computing. Distrib. Comput. **26**(4), 223–242 (2013)
18. Fraigniaud, P., Rajsbaum, S., Travers, C.: On the number of opinions needed for fault-tolerant run-time monitoring in distributed systems. In: Bonakdarpour, B., Smolka, S.A. (eds.) RV 2014. LNCS, vol. 8734, pp. 92–107. Springer, Cham (2014). doi:10.1007/978-3-319-11164-3_9
19. Göös, M., Suomela, J.: Locally checkable proofs. In: 30th ACM Symposium on Principles of Distributed Computing (PODC), pp. 159–168 (2011)
20. Korman, A., Kutten, S.: Distributed verification of minimum spanning trees. Distrib. Comput. **20**, 253–266 (2007)
21. Korman, A., Kutten, S., Masuzawa, T.: Fast and compact self stabilizing verification, computation, and fault detection of an MST. In: 30th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 311–320 (2011)

22. Korman, A., Kutten, S., Peleg, D.: Proof labeling schemes. Distrib. Comput. **22**(4), 215–233 (2010)
23. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press, Cambridge (1997)
24. Nash-Williams, C.S.A.: Edge-disjoint spanning trees of finite graphs. J. Lond. Math. Soc. **s1–36**(1), 445–450 (1961)
25. Nash-Williams, C.S.A.: Decomposition of finite graphs into forests. J. Lond. Math. Soc. **s1–39**(1), 12 (1964)
26. Patt-Shamir, B., Perry, M.: Proof-labeling schemes: broadcast, unicast and in between. CoRR, abs/1708.06947 (2017)
27. Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. Society for Industrial and Applied Mathematics, Philadelphia (2000)