# Network Verification via Routing Table Queries*

Evangelos Bampas[1], Davide Bilò[2], Guido Drovandi[3], Luciano Gualà[4], Ralf Klasing[1], and Guido Proietti[3,5]

[1] LaBRI, CNRS / INRIA / University of Bordeaux, Bordeaux, France**
[2] Dip. di Teorie e Ricerche dei Sistemi Culturali, University of Sassari, Italy
[3] Istituto di Analisi dei Sistemi ed Informatica, CNR, 00185 Rome, Italy
[4] Dipartimento di Matematica, University of Tor Vergata, Rome, Italy
[5] Dipartimento di Informatica, University of L'Aquila, L'Aquila, Italy

**Abstract.** We address the problem of *verifying* the accuracy of a map of a network by making as few measurements as possible on its nodes. This task can be formalized as an optimization problem that, given a graph $G = (V, E)$, and a *query model* specifying the information returned by a query at a node, asks for finding a *minimum-size* subset of nodes of $G$ to be queried so as to univocally identify $G$. This problem has been faced w.r.t. a couple of query models assuming that a node had some *global* knowledge about the network. Here, we propose a new query model based on the *local* knowledge a node instead usually has. Quite naturally, we assume that a query at a given node returns the associated *routing table*, i.e., a set of entries which provides, for each destination node, a corresponding (set of) first-hop node(s) along an underlying shortest path. First, we show that any network of $n$ nodes needs $\Omega(\log \log n)$ queries to be verified. Then, we prove that there is no $o(\log n)$-approximation algorithm for the problem, unless $\mathsf{P} = \mathsf{NP}$, even for networks of diameter 2. On the positive side, we provide an $O(\log n)$-approximation algorithm to verify a network of diameter 2, and we give exact polynomial-time algorithms for paths, trees, and cycles of even length.

## 1 Introduction

There is a growing interest about networks which are built and maintained by decentralized processes. In such a setting, it naturally arises the problem of discovering a map of the network or to verify whether a given map is accurate. A common approach to discover or to verify a map is to make some local measurement on a selected subset of nodes that – once collected – can be used to derive information about the whole network (see for instance [6,9]). A measurement on a node is usually costly, so it is natural to try to make as few measurements as possible.

These two tasks – that of *discovering* a map and that of *verifying* a given map – have been formalized as optimization problems and have been studied in

---

* Part of this work was done while the second author was visiting LaBRI-Bordeaux.

** Additional support by the ANR projects ALADDIN and IDEA and the INRIA project CEPAGE.

A. Kosowski and M. Yamashita (Eds.): SIROCCO 2011, LNCS 6796, pp. 270–281, 2011.

several papers. The idea is to model the network as a graph $G = (V, E)$, while a measurement at a given node can be seen as a unitary-cost *query* returning some piece of information about $G$. In the *discovery* problem, we want to design an on-line algorithm that selects a minimum-size subset of nodes $Q \subseteq V$ to be queried that allows to precisely map the entire graph, i.e., to settle all the edges and all the non-edges of $G$. The quality of the algorithm is measured by its competitive ratio, i.e., the ratio between the number of queries made by the algorithm (which does not know $G$) and the minimum number of queries which would be sufficient to discover the graph. On the other hand, in the off-line version of the problem, which is of interest for our paper, we are given a graph $G$, and we want to compute a minimum number of queries sufficient to discover $G$. This is known as the *verification* problem, and it has an interesting application counterpart, since it models the activity of verifying the accuracy of a given map associated with an underlying real network (on which the queries are actually done).

In the literature, two (main) query models have been studied. In the *all-shortest-paths* query model, a query of a node $q$ returns the subgraph of $G$ consisting of the union of all shortest paths between $q$ and every other node $v \in V$. A weaker notion of query is used in the *all-distances* query model, in which a query to a node $q$ returns all the distances in $G$ from $q$ to every other node $v \in V$. Notice that both models inherently require global knowledge/information about the network, hence a central problem for these query models is whether/how the information can be obtained locally (without preprocessing of the network). In this paper, we propose a query model that uses only *local* knowledge/information about the network. Quite naturally, we assume that a query at a given node $q$ returns the associated *routing table*, namely a set of entries which provides, for each destination node, a corresponding (set of) first-hop node(s) along an underlying shortest path. In the rest of the paper, this will be referred to as the *routing-table query model.*

*Previous work.* It turns out that the verification problem with the *all-shortest-paths* query model is equivalent to the problem of placing landmarks on a graph [14]. In this problem, we want to place landmarks on a subset of the nodes in such a way that every node is uniquely identified by the distance vector to the landmarks. Interestingly enough, the minimum number of landmarks to be placed is called the *metric dimension* of a graph [13]. The problem has been shown to be NP-hard in [8]. An explicit reduction from 3-SAT is given in [14] which also provides an $O(\log n)$-approximation algorithm ($n$ is the number of nodes) and an exact polynomial-time algorithm for trees. Subsequently, in [1], the authors prove that the problem is not $o(\log n)$ approximable, showing thus that the algorithm in [14] is the best possible in an asymptotic sense.

As far as the *all-distances* query model is concerned, the verification problem has been studied in [1] where the NP-hardness is proved and an algorithm with $O(\log n)$-approximation guarantee is provided. Other results in [1] include exact polynomial-time algorithms for trees, cycles and hypercubes. Problems close in spirit to the verification have been addressed in [2,4,5], while for the state of art about the discovery problem in both models, the reader is referred to [1,3,7].

*Our results.* Throughout the paper, we focus on the verification problem w.r.t the routing-table query model. We first show a lower bound of $\Omega(\log\log n)$ on the minimum number of queries needed to verify any graph with $n$ nodes. This is in contrast with the previous two query models for which certain classes of graphs can be verified with a constant number of queries, like paths and cycles. Our proof also implies a lower bound of $\Omega(n)$ on the number of queries needed to verify a path or a cycle. So, one can wonder whether every graph needs a linear number of queries to be verified. We provide a negative answer to this question by exhibiting a class of graphs that can be verified with $O(\log n)$ queries.

We then analyze the computational complexity of the problem. To this respect, although it remains open for general input graphs to establish whether the problem is in NPO, we are able to provide an $O(\log n)$-approximation algorithm to verify graphs of diameter 2. Moreover, we also show that this bound is asymptotically tight, unless P = NP. On the positive side, we provide exact polynomial-time algorithms to verify paths, trees and cycles of even length. Our result for trees is based on a characterization of a solution that can be used to reduce the problem to that of computing a minimum vertex cover of a certain class of graphs (for which a vertex cover can be found in polynomial time). The algorithm for cycles of even length shows a counterintuitive fact about the routing-table query model. Indeed, while a query in our model seems to obtain only local information about the graph, we show in the case of the cycle that the symmetry can be used to infer some knowledge about edges and non-edges that are far from queried nodes.

The paper is organized as follows. After giving some basic definitions in Section 2, we formally introduce our query model in Section 3. Section 4 is devoted to the lower bound of $\Omega(\log\log n)$ for any graph with $n$ nodes, while the results for graphs of diameter 2 are presented in Section 5. Then, in Section 6, we describe exact polynomial-time algorithms for classical topologies, and finally Section 7 concludes the paper. Due to space limitations, some of the proofs are omitted/sketched here, and will be given in the extended version of the paper.

## 2 Basic Definitions

Let $G = (V, E)$ be an undirected (simple) graph with $n$ vertices. We assume that vertices are distinguishable, i.e., they have different identifiers. If $(u, v) \notin E$, then we say that $(u, v)$ is a *non-edge* of $G$. For a graph $G$, we will also denote by $V(G)$ and $E(G)$ its set of vertices and its set of edges, respectively. For every vertex $v \in V$, let $N_G(v) := \{u \mid u \in V \setminus \{v\}, (u, v) \in E\}$ and let $N_G[v] = N_G(v) \cup \{v\}$. The *maximum degree* of $G$ is equal to $\max_{v \in V} |N_G(v)|$. Let $U \subseteq V$ be a set of vertices. We denote by $G[U]$ the graph with $V(G[U]) = U$ and $E(G[U]) = \{(u, v) \mid u, v \in U, (u, v) \in E\}$. Let $F \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$. We denote by $G + F$ (resp., $G - F$) the graph on $V$ with edge set $E \cup F$ (resp., $E \setminus F$). When $F = \{e\}$ we will denote $G + \{e\}$ (resp., $G - \{e\}$) by $G + e$ (resp., $G - e$). For two graphs $G_1$ and $G_2$, we denote by $G_1 \cup G_2$ the graph with $V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$ and $E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$. We denote by $d_G(u,v)$ the *distance* in $G$ from $u$ to $v$. The *diameter* of $G$ is equal to $\max_{u,v \in V} d_G(u,v)$.

Let $\mathtt{query}_G$ be a *query model*, that is, a function from vertices of $G$ to some information about $G$. Let $Q \subseteq V$. We denote by $\mathtt{query}_G(Q) = \{\mathtt{query}_G(q) \mid q \in Q\}$. Moreover, we say that $Q$ *verifies* edge (resp., non-edge) $(u,v)$ of $G$ iff for every graph $G' = (V, E')$ with $\mathtt{query}_G(Q) = \mathtt{query}_{G'}(Q)$ we have that $(u,v) \in E'$ (resp., $(u,v) \notin E'$). Finally, $Q$ *verifies* $G$ iff for every $G' = (V, E')$ with $E \neq E'$ we have that $\mathtt{query}_G(Q) \neq \mathtt{query}_{G'}(Q)$. This implies that $Q$ verifies $G$ iff $Q$ verifies every edge and every non-edge of $G$. Clearly, we have that if $Q \subseteq V$ verifies $G$, then for every $q \in V$, $Q \cup \{q\}$ verifies $G$.

Given an undirected graph $G = (V, E)$, the *Network Verification Problem w.r.t. query model* $\mathtt{query}_G$ is the optimization problem of finding a *minimum-size* subset $Q \subseteq V$ that verifies $G$ w.r.t. query model $\mathtt{query}_G$.

## 3 The Routing-Table Query Model

For a given vertex $q \in V$, we denote by $\mathtt{table}_G(q)$ the *routing table* of $q$ in $G$, i.e.,

$$\mathtt{table}_G(q) = \Big\{\langle u, v\rangle \mid u, v \in V \setminus \{q\} \wedge (q, v) \in E \wedge d_G(u, v) + 1 = d_G(q, u)\Big\}.$$

A pair $\langle u, v\rangle \in \mathtt{table}_G(q)$ means that there exists a shortest path from $q$ to $u$ whose first hop is vertex $v$. The *routing-table query model* is the model in which $\mathtt{query}_G(q) = \mathtt{table}_G(q)$, for every $q \in V$. In the rest of the paper, we will denote by $\mathtt{T}^q_G(v) = \big\{u \in V \mid \langle u, v\rangle \in \mathtt{table}_G(q)\big\}$. Clearly, for every $v \in V$ we have that $Q = V \setminus \{v\}$ verifies $G$ w.r.t. the routing table query model, as any $q \in Q$ verifies all edges and non-edges of $G$ of the form $(q, u)$, for any $u \in V \setminus \{q\}$. Notice also that if $G$ is a clique, then this is optimal.

The following fact is easy to prove:

**Fact 1.** *Let $q$ and $u$ be two vertices of $G$ such that $(q, u) \in E$. For every $v \in \mathtt{T}^q_G(u)$, there is a shortest path between $q$ and $v$ using edge $(q, u)$ and using only some of the vertices in $\mathtt{T}^q_G(u)$. Moreover, if for every other $u' \neq u$, we have that $v \notin \mathtt{T}^q_G(u')$, then all the shortest paths between $q$ and $v$ must use edge $(q, u)$ and must use only vertices in $\mathtt{T}^q_G(u)$.*

As a consequence of the above fact, we are now able to give some easy-to-check conditions which are sufficient to verify a given edge (respectively, non-edge) w.r.t. routing-table query model. Unfortunately, these conditions are not necessary, and so it remains open to establish whether the problem is in $\mathsf{NPO}$.

**Proposition 1.** *Let $(u, v)$ be an edge of $G$. Let $q$ be such that $\mathtt{T}^q_G(u) = \{v\}$. Then, $\{q\}$ verifies edge $(u, v)$.*

**Proposition 2.** *Let $(u, v)$ be an edge of $G$. Let $q$ be a neighbor of $u$ and $q'$ a neighbor of $v$, respectively. If $\mathtt{T}^{q'}_G(v) \cap \mathtt{T}^q_G(u) = \{u, v\}$, then $\{q, q'\}$ verifies the edge $(u, v)$.*

**Proposition 3.** *Let $(u,v)$ be a non-edge of $G$. Let $q \in V \setminus \{u,v\}$ be such that $(q,u) \in E$, $(q,v) \notin E$. If $v \notin \mathtt{T}^q_G(u)$, then $\{q\}$ verifies the non-edge $(u,v)$.*

**Proposition 4.** *Let $(u,v)$ be a non-edge of $G=(V,E)$ and let $q,q' \in V \setminus \{u,v\}$ be two distinct vertices such that $(q,q') \in E$. If there exists $w \in V$ such that $v \notin \mathtt{T}^q_G(w)$, $u \in \mathtt{T}^q_G(w)$, $v \in \mathtt{T}^q_G(q')$, and $u \in \mathtt{T}^{q'}_G(q)$, then $\{q,q'\}$ verifies non-edge $(u,v)$.*

*Proof.* For the sake of contradiction, assume there exists a graph $G'=(V,E')$ satisfying the hypothesis of the claim such that $(u,v) \in E'$. This implies that $|d_{G'}(z,u) - d_{G'}(z,v)| \le 1$ for every vertex $z \in V$. As $v \notin \mathtt{T}^q_{G'}(w)$ and $u \in \mathtt{T}^q_G(w)$, we have that $d_{G'}(q,v) \le d_{G'}(q,u)$. Moreover, as $u \in \mathtt{T}^{q'}_{G'}(q)$ and $v \in \mathtt{T}^q_{G'}(q')$, we have that $d_{G'}(q',u) = d_{G'}(q,u)+1$ and $d_{G'}(q',v) = d_{G'}(q,v)-1$. As a consequence, $d_{G'}(q',v) = d_{G'}(q,v)-1 \le d_{G'}(q,u)-1 = d_{G'}(q',u)-2$, which implies $|d_{G'}(q',u) - d_{G'}(q',v)| \ge 2$, contradicting the assumption that $|d_{G'}(q',u) - d_{G'}(q',v)| \le 1$. This completes the proof. □

Before ending this section, we provide some connections between the routing-table query model and the all-shortest-paths query model, which will be useful in the following sections. First, we recall the formal definition of the all-shortest-paths query model. For two given vertices $u,v \in V$, let $\Pi_G(u,v)$ denote the graph obtained by the union of all shortest paths in $G$ between $u$ and $v$. For a given vertex $q \in V$, we denote by $\mathtt{asp}_G(q) = \bigcup_{u \in V} \Pi_G(q,u)$. The *all-shortest-paths query model* is the model in which $\mathtt{query}_G(q) = \mathtt{asp}_G(q)$, for every $q \in V$.

**Lemma 1 ([1]).** *A set $Q \subseteq V$ verifies a graph $G=(V,E)$ w.r.t. the all-shortest-paths query model iff, for every $u,v \in V$, with $u \neq v$, there exists a vertex $q \in Q$ such that $|d_G(q,u) - d_G(q,v)| \ge 1$.*

As from $\mathtt{asp}_G(q)$ we can easily construct $\mathtt{table}_G(q)$, the routing-table query model is weaker than the all-shortest-paths query model. More formally,

**Proposition 5.** *If $Q \subseteq V$ verifies $G$ w.r.t. the routing-table query model, then $Q$ verifies $G$ w.r.t. the all-shortest-paths query model.*

One can wonder whether the routing-table query model is always much weaker than the all-shortest-paths query model. The following lemma, which we will use also in the rest of the paper, shows that this is not the case for some class of graphs.

**Proposition 6.** *Let $G=(V,E)$ be a graph containing a vertex $s$ which is adjacent to all other vertices of $G$. Let $Q \subseteq V$. If $Q$ verifies $G$ w.r.t. the all-shortest-paths query model, then $Q \cup \{s\}$ verifies $G$ w.r.t. the routing-table query model.*

*Proof.* Let $u,v \in V, u \neq v$ and $u,v \notin Q$. As $Q$ verifies $G$ w.r.t. the all-shortest-paths query model, then Lemma 1 implies that $|d_G(q,u) - d_G(q,v)| \ge 1$, for some $q \in Q$. W.l.o.g., let $d_G(q,u) < d_G(q,v)$. Now, consider the routing-table query model. After making the query at $s$, we know that every vertex is at distance 1 from $s$. Thus, the distance between any pair of distinct vertices of $G$ can be either 1 or 2. Since $\langle u,u\rangle \in \texttt{table}_G(q)$ whilst $\langle v,v\rangle \notin \texttt{table}_G(q)$, we know that $d_G(q,u)=1$ whilst $d_G(q,v)=2$. Therefore, $(u,v)$ is an edge of $G$ iff $\langle v,u\rangle \in \texttt{table}_G(q)$ (and thus, $(u,v)$ is a non-edge of $G$ iff $\langle v,u\rangle \notin \texttt{table}_G(q)$). Hence, $Q \cup \{s\}$ verifies $G$ w.r.t. the routing-table query model. □

## 4 Lower Bounds on the Size of Feasible Solutions

In this section, we show lower bounds on the minimum number of queries needed to verify any graph $G$ of $n$ vertices w.r.t. the routing-table query model, as well as improved (linear) lower bounds for paths and cycles.

We begin by showing that $\Omega(\log\log n)$ queries are necessary to verify a graph $G$ of $n$ vertices. In [1], the authors proved that $\log_3 \Delta$ queries are necessary to verify a graph $G$ of maximum degree equal to $\Delta$ w.r.t. the all-shortest-paths query model. Therefore, by Proposition 5, we have that

**Corollary 1.** *Let $G$ be a graph of maximum degree equal to $\Delta$ and let $Q$ be a query set that verifies $G$ w.r.t. the routing-table query model. Then $|Q| \geq \log_3 \Delta$.*

In what follows, we prove a lower bound of $\Omega\big(\frac{\log n}{\Delta}\big)$ on the minimum number of queries needed to verify a graph $G$ with $n$ vertices and maximum degree equal to $\Delta$ w.r.t. the routing-table query model, thus obtaining a lower bound of $\max\{\log_3 \Delta, \Omega\big(\frac{\log n}{\Delta}\big)\} = \Omega(\log\log n)$ on the minimum number of queries needed to verify any graph $G$ of $n$ vertices w.r.t. the routing-table query model. For any $q, v \in V$, let

$$\texttt{group}^q_G(v) := \begin{cases} \{w \in V \mid v \in \texttt{T}^q_G(w)\} & \text{if } v \notin N_G[q];\\ \emptyset & \text{otherwise.}\end{cases}$$

The lower bound of $\Omega\big(\frac{\log n}{\Delta}\big)$ hinges on the following necessary condition

**Proposition 7.** *If $Q \subseteq V$ verifies $G$, then, $\forall u, v \in V, u \neq v$, one of the following conditions is satisfied:*

*(i) $u \in N_G[q]$ or $v \in N_G[q]$, for some $q \in Q$;*
*(ii) $\exists q \in Q$ such that $\texttt{group}^q_G(u) \neq \texttt{group}^q_G(v)$.*

*Proof.* For the sake of contradiction, assume that $Q$ verifies $G$ but none of the conditions (i) and (ii) is satisfied. We divide the proof into two cases.

In the first case, we have that $u$ and $v$ are *twin* vertices, i.e., the identity function from $V$ to $V$ is an isomorphism for $G$ and the graph obtained from $G$ by swapping the role of $u$ and $v$. As (i) is not satisfied, we have that $u, v \notin Q$. Therefore, $d_G(q,u) = d_G(q,v)$ for every $q \in Q$. Thus, Proposition 5 and Lemma 1 imply that $Q$ cannot verify $G$.

In the second case, we have that $u$ and $v$ are not twin vertices. Consider the graph $G'$ obtained from $G$ by swapping the role of $u$ and $v$. Clearly, for every $q \in Q$, and for every vertex $x \in V, x \neq u, v$, $\langle x, w\rangle \in \texttt{table}_G(q)$ iff $\langle x, w\rangle \in$

$\texttt{table}_{G'}(q)$, since $w \neq u, v$ as condition (i) does not hold. Moreover, by definition of $G'$ and because (i) does not hold, $u \in \texttt{T}^q_{G'}(w)$ iff $v \in \texttt{T}^q_{G'}(w)$ and $v \in \texttt{T}^q_G(w)$ iff $u \in \texttt{T}^q_{G'}(w)$, for every $q \in Q$. Since (ii) does not hold, then for every $q \in Q$, $u \in \texttt{T}^q_G(w)$ iff $v \in \texttt{T}^q_G(w)$. As a consequence, for every $q \in Q$, $u \in \texttt{T}^q_G(w)$ iff $u \in \texttt{T}^q_{G'}(w)$ and $v \in \texttt{T}^q_G(w)$ iff $v \in \texttt{T}^q_{G'}(w)$. Therefore, $\texttt{table}_G(q) = \texttt{table}_{G'}(q)$ for every $q \in Q$. Thus, $Q$ cannot verify $G$. □

We can prove the following.

**Lemma 2.** *Let $G$ be a graph with $n$ vertices of maximum degree equal to $\Delta$ and let $Q$ be a set of queries that verifies $G$ w.r.t. the routing-table query model. Then $|Q| \geq \Omega\big(\frac{\log n}{\Delta}\big)$.*

*Proof.* Let $Q = \{q_1, \ldots, q_h\}$ be a minimum cardinality set of queries that verifies $G$ w.r.t. the routing-table query model and let $V' = V \setminus \bigcup_{q \in Q} N_G[q]$. Since $G$ has maximum degree equal to $\Delta$, we have that $|V'| \geq n - |Q|(\Delta + 1)$. Moreover, as $\texttt{group}^q_G(v) \subseteq N_G(q)$, for every $v \in V'$ and for every $q \in Q$, we have that $\texttt{group}^q_G(v)$ is an element of the power set $2^{\Delta}$ . As a consequence, $\langle \texttt{group}^{q_1}_G(v), \ldots, \texttt{group}^{q_h}_G(v) \rangle$ is an element of the power set $(2^{\Delta})^{|Q|} = 2^{|Q|\Delta}$. Since condition (ii) of Proposition 7 implies that $\langle \texttt{group}^{q_1}_G(v), \ldots, \texttt{group}^{q_h}_G(v) \rangle \neq \langle \texttt{group}^{q_1}_G(u), \ldots, \texttt{group}^{q_h}_G(u) \rangle$ for every two distinct vertices $u, v \in V'$, we have that

$$2^{|Q|\Delta} \geq |V'| \geq n - |Q|(\Delta + 1)$$

holds. Hence, $|Q| = \Omega\big(\frac{\log n}{\Delta}\big)$. □

By combining the lower bound in Corollary 1 with the one in Lemma 2 we obtain

**Theorem 1.** *Let $G$ be a graph of $n$ vertices and let $Q$ be a query set that verifies $G$ w.r.t. the routing-table query model. Then $|Q| = \Omega(\log \log n)$.*

We point out that a direct application of Proposition 7 implies linear lower bounds for paths and cycles (unlike in the all-shortest-paths query model, for which a constant number of queries suffices). More formally,

**Corollary 2.** *Let $G$ be a graph of $n$ vertices and let $Q$ be a minimum cardinality set of queries that verifies $G$ w.r.t. the routing-table query model. We have that*

1. *$|Q| \geq \left\lfloor \frac{n}{4} \right\rfloor$ if $G$ is a path;*
2. *$|Q| \geq \left\lfloor \frac{n}{8} \right\rfloor$ if $G$ is a cycle.*

*Proof.* For paths, by Proposition 7, at least one vertex of every subpath of four consecutive vertices has to be contained in $Q$. The proof for cycles is omitted. □

In Section 6, we provide an improved (tight) lower bound for paths.

Due to the results of Corollary 2, one can wonder whether every graph needs a linear number of queries to be verified. We provide a negative answer to this question by exhibiting a class of graphs that can be verified with $O(\log n)$ queries.

Consider any graph $G'$ of $n'$ vertices $u_0, \ldots, u_{n'-1}$. We build $G$ as follows. $G$ contains a copy of $G'$ plus $1 + \lceil \log n' \rceil$ vertices $s, q_1, \ldots, q_{\lceil \log n' \rceil}$. Vertex $u_i$ is adjacent to vertex $q_j$ iff the $j$-th bit of the binary representation of $i$ is equal to 1. Vertex $s$ is adjacent to all the other vertices of $G$. Let $Q = \{s, q_1, \ldots, q_{\lceil \log n' \rceil}\}$. We now argue that $Q$ verifies $G$ w.r.t. the all-shortest-path query model. Indeed, $Q$ verifies all edges and non-edges incident to the vertices in $Q$. Moreover, for every $u_i, u_{i'}$ with $i \neq i'$, there exists at least one bit, say the $j$-th, in which the binary representation of $i$ differs from the binary representation of $i'$. This implies that $|d_G(q_j, u_i) - d_G(q_j, u_{i'})| \geq 1$. Thus, Lemma 1 implies that $Q$ verifies $G$ w.r.t. the all-shortest-paths query model. Finally, as $s$ is adjacent to all vertices of the graph, by Proposition 6 we have that $Q$ verifies $G$ w.r.t the routing-table query model.

## 5 Verifying Graphs of Diameter 2

Even though the problem of determining whether the Network Verification Problem w.r.t. the routing-table query model is in NPO is open, in this section, we first show the existence of a polynomial-time algorithm that computes a set of queries $Q$ that verifies a graph $G$ of diameter equal to 2 w.r.t. the routing-table query model such that $|Q|$ is within an $O(\log n)$ (multiplicative) factor from the size of any optimal solution. Furthermore, we also show that this result is asymptotically best possible. Indeed, for graphs of diameter equal to 2, we prove that no polynomial time algorithm can compute a set of queries that verifies the graph w.r.t. the routing-table query model whose size is within an $o(\log n)$ (multiplicative) factor from the size of any optimal solution.

To describe our algorithm, we need to introduce some definitions. Let $G = (V, E)$ be a graph. A set $U \subseteq V$ is a *locating-dominating code* of $G$ iff (i) $N_G[v] \cap U \neq \emptyset$ for every $v \in V$ and (ii) $N_G(v) \cap U \neq N_G(u) \cap U$ for every $u, v \in V \setminus U, u \neq v$. A set $U \subseteq V$ is a *connected* locating-dominating code iff $U$ is a locating-dominating code and $G[U]$ is a connected graph. The optimization problem of computing a minimum cardinality locating-dominating code of a graph $G$ of $n$ vertices can be approximated within a factor of $O(\log n)$ and this ratio is asymptotically tight [10,15].

**Lemma 3.** *Let $G = (V, E)$ be a graph of diameter equal to 2, let $U^*$ be a minimum cardinality locating-dominating code of $G$, and let $Q \subseteq V$ be a set of vertices that verifies $G$ w.r.t. the routing-table query model. Then $|Q| \geq |U^*| - 1$.*

*Proof.* Let $u, v \in V \setminus Q$, with $u \neq v$. By Proposition 5 we have that if $Q$ verifies $G$, then $|d_G(q, u) - d_G(q, v)| \geq 1$, for some $q \in Q$. As $G$ has diameter equal to 2, we either have that $d_G(q, u) = 1$ and $d_G(q, v) = 2$ or $d_G(q, u) = 2$ and $d_G(q, v) = 1$. As this has to be true for every two distinct vertices $u, v \in V \setminus Q$, it follows that there exists at most one vertex, say $\bar{v}$ such that $d_G(q, \bar{v}) = 2$ for every $q \in Q$. As a consequence, $Q \cup \{\bar{v}\}$ is a locating-dominating code of $G$. Thus, $|Q| + 1 \geq |U^*|$. □

**Theorem 2.** *Let $G = (V, E)$ be a graph of diameter equal to 2 and let $Q^*$ be a minimum cardinality set of queries that verifies $G$ w.r.t. the routing-table query model. There exists a polynomial-time algorithm that computes a set $Q$ that verifies $G$ w.r.t. the routing-table query model such that $\frac{|Q|}{|Q^*|} = O(\log n)$.*

*Proof.* We prove that any connected locating-dominating code of $G$ verifies $G$. Observe that this immediately implies the claim. Indeed, let $U^*$ be a minimum cardinality locating-dominating code of $G$. As $U^*$ is also a dominating set of $G$, it is easy to construct a connected locating-dominating code $U$ of $G$ such that $U^* \subseteq U$ and $|U| \leq O(|U^*|)$ (see also [11]). Therefore, thanks to the $O(\log n)$-approximation algorithm for computing a locating-dominating code of $G$ (see [10]), we can also compute a connected locating-dominating code $Q$ of $G$ such that $\frac{|Q|}{|U^*|} = O(\log n)$. Thus, Lemma 3 implies $\frac{|Q|}{|Q^*|} \leq \frac{|Q|}{|U^*|-1} = O(\log n)$.

Let $Q$ be a connected locating-dominating code of $G$ and consider two distinct vertices $u$ and $v$ of $G$ such that $u, v \notin Q$. As $Q$ is a locating-dominating code, then there exists a vertex $q \in Q$ such that, w.l.o.g., $q \in N_G(u)$ and $q \notin N_G(v)$, i.e, $\langle u, u \rangle \in \mathtt{table}_G(q)$ and $\langle v, v \rangle \notin \mathtt{table}_G(q)$. This implies that $d_G(q, u) = 1$ and $d_G(q, v) \geq 2$.

If $(u, v)$ is a non-edge, then as the diameter of $G$ is equal to 2, we have that $v \notin \mathtt{T}^q_G(u)$. Therefore, by Proposition 3, we have that $Q$ verifies non-edge $(u, v)$.

If $(u, v)$ is an edge, then we have that $v \in \mathtt{T}^q_G(u)$. Let $q'$ be a vertex of $Q$ be such that $q' \in N_G(v)$, i.e., $\langle v, v \rangle \in \mathtt{table}_G(q')$. If $d_G(q, q') = 1$, then we have that $\langle v, q' \rangle \in \mathtt{table}_G(q)$ and thus we know that $d_G(q, v) = 2$. Therefore, $v \in \mathtt{T}^q_G(u)$ implies that $(u, v)$ is an edge of $G$. Consider the case $d_G(q, q') \geq 2$ for every vertex $q' \in Q$ such that $q' \in N_G(v)$. Since $G$ has diameter equal to 2, we have that $d_G(q, q') = 2$. Moreover, as $Q$ is a connected locating-dominating code, there is a vertex $q'' \in Q$ such that $d_G(q, q'') = d_G(q', q'') = 1$. As $d_G(\bar{q}, v) = 2$ for every vertex $\bar{q} \in Q$ such that $d_G(q, \bar{q}) = 1$, we have that $\langle v, \bar{q} \rangle \notin \mathtt{table}_G(q)$. Furthermore, after querying $\bar{q}$ and all the $q' \in Q$ such that $q' \in N_G(v)$, we know that $d_G(\bar{q}, v) = 2$. As a consequence, $\langle v, \bar{q} \rangle \notin \mathtt{table}_G(q)$ implies that $d_G(q, v) \leq d_G(\bar{q}, v) + 1 - 1 \leq 2$. As $d_G(q, v) \geq 2$, we have that $d_G(q, v) = 2$. Furthermore, as $d_G(q, v) = 1 + d_G(u, v)$, it follows that $(u, v)$ is an edge of $G$. □

We observe that the result of Theorem 2 is asymptotically tight due to the following

**Theorem 3.** *There exists a class of graphs $\mathcal{G}$ and a constant $c > 0$ such that, for every $G \in \mathcal{G}$ of $n$ vertices and for every $c' \leq c$, unless $\mathsf{P} = \mathsf{NP}$ no polynomial-time algorithm computes a set $Q$ of queries that verifies $G$ w.r.t. the routing-table query model of size $|Q| \leq c'|Q^*| \log n$, where $Q^*$ is a minimum cardinality set of queries that verifies $G$.*

*Proof.* In [1], the authors proved that the Network Verification Problem w.r.t. the all-shortest-paths query model has a lower bound of $\Omega(\log n)$ on its approximability ratio, unless $\mathsf{P} = \mathsf{NP}$. Their reduction consists of a graph $G$ having a vertex which is adjacent to all other vertices of $G$. The claim now follows as a consequence of Proposition 5 and Proposition 6. □

# 6 Optimal Algorithms for Classical Topologies

In this section, we show that the Network Verification Problem w.r.t. the routing-table query model can be solved optimally in linear time on paths and trees. Besides that, for cycles of even length we are able to build an optimal query set of size $2\lfloor n/6 \rfloor + \frac{n}{2} \bmod 3$. Due to space limitations this result is omitted, but it is worth noting here that our approach heavily relies on the existence of antipodal nodes in the cycle, and so it is not easily extendible to cycles of odd length.

## 6.1 Paths

It is clear that a path of 2 vertices can be verified by querying any of the two vertices. Let $P_n$ be a path of $n \geq 3$ vertices. In what follows we show that $2\lfloor \frac{n}{4} \rfloor + \lfloor \frac{n \bmod 4}{3} \rfloor$ queries are necessary to verify $P_n$. We also show how to verify $P_n$ with $2\lfloor \frac{n}{4} \rfloor + \lfloor \frac{n \bmod 4}{3} \rfloor$ queries. We number the vertices of $P_n$ from 1 to $n$ by traversing the path from one endvertex to the other one. We have that

**Lemma 4.** *Let $Q \subseteq V(P_n)$. $Q$ verifies $P_n$ iff for every $i = 1, \ldots, n-2$, $i \in Q$ or $i + 2 \in Q$.*

*Proof.* Let $Q$ be a set of vertices that verifies $G$ and, for the sake of contradiction, assume that there exists $i \in \{1, \ldots, n-2\}$ such that $i, i+2 \notin Q$. It is easy to verify that $\mathtt{table}_G(j) = \mathtt{table}_{G+(i,i+2)}(j)$ for every $j = 1, \ldots, n, j \neq i, i+2$. Indeed, for every $k = 1, \ldots, n$, with $k \neq j$, we have that $\langle k, j \pm 1 \rangle \in \mathtt{table}_G(j)$ iff $\langle k, j \pm 1 \rangle \in \mathtt{table}_{G+(i,i+2)}(j)$.

Now, let $Q$ be a set of vertices such that $i \in Q$ or $i + 2 \in Q$, for every $i = 1, \ldots, n-2$. We show that $Q$ verifies $G$. Consider any vertex $i$ not in $Q$. We prove that $Q$ verifies all the edges and non-edges of the form $(i, j)$, for every $j = 1, \ldots, n$, $j \neq i$.

If $i \neq 1, n-1, n$, as $i+1 \in Q$ or $\{i-1, i+2\} \subseteq Q$, then by Proposition 2, we have that $Q$ verifies all edges $(i, i+1)$ with $2 \leq i \leq n-2$. If $1 \in Q$ or $2 \in Q$, then also edge $(1, 2)$ is verified. If $1, 2 \notin Q$, then $3 \in Q$ and thus, by Proposition 1, we have that $Q$ also verifies edge $(1, 2)$. The proof for the edge $(n-1, n)$ is similar to the proof of edge $(1, 2)$.

Concerning the non-edges of $G$, let $(i, j)$ with $j \geq i + 3$ be any non-edge of $G$ such that $j \notin Q$. Notice that there exists $i < k < j - 1$ such $k, k+1 \in Q$. By Proposition 4, we have that $\{k, k+1\}$ verifies non-edge $(i, j)$. □

Thanks to Lemma 4, we can reduce the problem of verifying $P_n$ to the problem of finding a *minimum vertex cover* on paths, a problem that can be clearly solved in linear time.[1] Indeed, consider the graph $G'$ with $V(G') = V(P_n)$ such that

[1] Assume the path contains $n$ vertices numbered from 1 to $n$ by traversing the path from one endvertex to the other one. The set $X = \{i \mid 1 \leq i \leq n, i \text{ is even}\}$ of $\lfloor n/2 \rfloor$ vertices is a vertex cover of the path. To see that it is minimum, first of all observe that $|X|$ is equal to the size of $\{(i-1, i) \mid 1 \leq i \leq n, i \text{ is even}\}$, a maximum matching of the path. Next, use the well-know König-Egerváry theorem stating that the size of any vertex cover of a graph $G$ is lower bounded by the size of any matching of $G$.

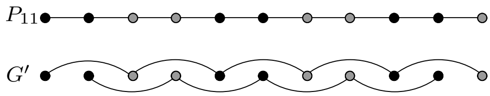


**Fig. 1.** An example of an optimal query set for $P_{11}$. Graph $G'$ on $V(P_{11})$ contains an edge between two vertices iff their distance in $P_{11}$ is 2. The set of gray vertices is a minimum vertex cover of $G'$ as well as a minimum-size set of queries that verifies $P_{11}$.

there exists edge $(i, j)$ iff $|i - j| = 2$ (i.e., either $j = i - 2$ or $j = i + 2$). We have that $Q$ verifies $P_n$ iff $Q$ is a vertex cover of $G'$.

Observe that the graph $G'$ is a forest of two paths, one containing all the $\lceil \frac{n}{2} \rceil$ odd vertices, and the other one containing all the $\lfloor \frac{n}{2} \rfloor$ even vertices (see also Figure 1). As the minimum cardinality vertex cover of a path of $k$ vertices is $\lfloor \frac{k}{2} \rfloor$, we have that $\lfloor \frac{n}{4} \rfloor + \lfloor \frac{\lceil \frac{n}{2} \rceil}{2} \rfloor = 2\lfloor \frac{n}{4} \rfloor + \lfloor \frac{n \bmod 4}{3} \rfloor$ vertices are necessary to verify $P_n$. Moreover, the set $\{4i \mid i = 1, \ldots, \lfloor \frac{n}{4} \rfloor\} \cup \{4i - 1 \mid i = 1, \ldots, \lfloor \frac{n}{4} \rfloor + \lfloor \frac{n \bmod 4}{3} \rfloor\}$ of $2\lfloor \frac{n}{4} \rfloor + \lfloor \frac{n \bmod 4}{3} \rfloor$ vertices verifies $P_n$ (see Figure 1). Thus, we have the following

**Theorem 4.** *There exists a linear-time algorithm that solves optimally the Network Verification Problem w.r.t. the routing-table query model on paths.*

### 6.2 Trees

We now extend the above algorithm for trees. Let $T$ be a tree of $n$ vertices. Observe that the proof of Lemma 4 can be easily extended to prove the following.

**Lemma 5.** *Let $Q \subseteq V(T)$. $Q$ verifies $T$ iff for every path $P$ in $T$, with $|V(P)| \geq 3$, $Q \cap V(P)$ verifies $P$.*

Then, the following can be proven

**Theorem 5.** *There exists a linear-time algorithm that solves optimally the Network Verification Problem w.r.t. the routing-table query model on trees.*

*Sketch of proof.* Thanks to Lemma 4 and 5, it can be shown that the problem of verifying $T$ reduces to finding a minimum vertex cover of $G' = (V(T), \{(u, v) \mid u, v \in V(T), d_T(u, v) = 2\})$. Graph $G'$ consists of two connected components, each of which is a *block graph* [12]. For such special graph we can provide a linear-time algorithm to compute a minimum vertex cover, and the claim follows. □

## 7 Conclusions

In this paper, we addressed the problem of verifying a graph w.r.t. the newly defined routing-table query model. On the one hand, we showed that the problem is NP-hard to approximate within $o(\log n)$ (which is tight for graphs of diameter 2), and on the other hand that it can be solved optimally in linear time for some basic network topologies.

We argued that our query model is much closer to reality than the previously used all-shortest-paths and all-distances query models, as it relies on local information that can be gathered by simply exploring the routing tables of the nodes of a given network. In practice, however, routing tables could contain much more information than the one we used in defining our query model (e.g., the distance to the destination nodes), or they might have a bounded number of next-hop entries for each specific destination node. Thus, we plan in the future to investigate corresponding variants of the introduced model. Moreover, for the presented query model and its envisioned variants, establishing whether the network verification problem is in NPO is a challenging research task.

## References


1. Beerliova, Z., Eberhard, F., Erlebach, T., Hall, A., Hoffman, M., Mihal'ák, M., Ram, S.: Network discovery and verification. IEEE Journal on Selected Areas in Communications 24(12), 2168–2181 (2006)
2. Bejerano, Y., Rastogi, M.: Rubust monitoring of link delays and faults in IP networks. In: 22nd IEEE Int. Conf. on Comp. Comm (INFOCOM 2003), pp. 134–144 (2003)
3. Bilò, D., Erlebach, T., Mihal'ák, M., Widmayer, P.: Discovery of network properties with all-shortest-paths queries. Theoretical Computer Science 411(14-15), 1626–1637 (2010)
4. Bshouty, N.H., Mazzawi, H.: Reconstructing weighted graphs with minimal query complexity. Theoretical Computer Science 412(19), 1782–1790 (2011)
5. Choi, S.-S., Kim, J.H.: Optimal query complexity bounds for finding graphs. Artificial Intelligence 174(9-10), 551–569 (2010)
6. Dall'Asta, L., Alvarez-Hamelin, J.I., Barrat, A., Vázquez, A., Vespignani, A.: Exploring networks with traceroute-like probes: Theory and simulations. Theoretical Computer Science 355(1), 6–24 (2006)
7. Erlebach, T., Hall, A., Mihal'ák, M.: Approximate Discovery of Random Graphs. In: Hromkovič, J., Královič, R., Nunkesser, M., Widmayer, P. (eds.) SAGA 2007. LNCS, vol. 4665, pp. 82–92. Springer, Heidelberg (2007)
8. Garey, M.R., Johnson, D.: Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco (1979)
9. Govindan, R., Tangmunarunkit, H.: Heuristics for Internet map discovery. In: 19th IEEE Int. Conf. on Comp. Comm (INFOCOM 2000), pp. 1371–1380 (2000)
10. Gravier, S., Klasing, R., Moncel, J.: Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs. Algorithmic Operations Research 3, 43–50 (2008)
11. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. Algorithmica 20, 374–387 (1998)
12. Harary, F.: A characterization of block graphs. Canad. Math. Bull. 6(1), 1–6 (1963)
13. Harary, F., Melter, R.: The metric dimension of a graph. Ars Combinatoria, 191–195 (1976)
14. Khuller, S., Raghavachari, B., Rosenfeld, A.: Landmarks in graphs. Discrete Applied Mathematics 70, 217–229 (1996)
15. Suomela, J.: Approximability of identifying codes and locating-dominating codes. Information Processing Letters 103(1), 28–33 (2007)