Tomographic Node Placement Strategies and the Impact of the Routing Model

Yvonne-Anne Pignolet, Stefan Schmid, Gilles Trédan

Tomographic Node Placement Strategies and the Impact of the Routing Model

Yvonne-Anne Pignolet, Stefan Schmid, Gilles Tredan

March 1, 2018

Abstract

Fault-tolerant computer networks rely on mechanisms supporting the fast detection of link failures. Tomographic techniques can be used to implement such mechanisms at low cost: it is often sufficient to deploy a small number of tomography nodes exchanging probe messages along paths between them and detect link failures based on these messages. Our paper studies a practically relevant aspect of network tomography: the impact of the routing model. While the relevance of the routing model on path diversity and hence tomography cost is obvious and well-known on an anecdotal level, we lack an analytical framework to quantify the influence of different routing models (such as destination-based routing) exists. This paper fills this gap and introduces a formal model for asymmetric network tomography and a taxonomy of path routing models. This facilitates algorithmic reasoning about tomographic placement problems and quantifying the difference between routing models. In particular, we provide optimal and near-optimal algorithms to deploy a minimal number of asymmetric and symmetric tomography nodes for basic network topologies (modelled as graphs) under different routing model classes. Interestingly, we find that in many cases routing according to a more restrictive routing model gives better results: compared to a more general routing model, computing a good placement is algorithmically more tractable and does not entail high monitoring costs, a desirable trade-off in practice.

1 Introduction

Computer networks often constitute a critical infrastructure and have to meet strict requirements in terms of availability. Accordingly, modern computer networks typically support robust routing and fast failover: upon a link failure, traffic is quickly rerouted along an alternative path. For instance, MPLS networks include different link and path protection schemes [1], and OpenFlow networks support conditional rules for inband local fast failover [5].

A crucial prerequisite for any resilient routing network is the ability to *detect* link failures. If not supported directly by the network itself, a dedicated monitoring infrastructure needs to be set up to actively check the health status of the network. Network tomography is a well-known approach to implement such a monitoring infrastructure at low cost: Rather than deploying and operating monitors at all nodes in a network, tomographic techniques can be used to probe paths only between a small number of tomography nodes.

Given the appealing properties of network tomography, tomographic techniques in general and the placement (deployment) of tomography nodes to monitor (multiple) links between them, have been studied intensively over the last years, in various settings. The classic optimization problem is to *minimize* the number of deployed tomography nodes: a tomographic infrastructure involves the development, installation, debugging, operation, and maintenance of specialized software/hardware on each tomography node. Usually in the literature, tomography nodes play a symmetric role: they serve both as sender and receiver of probes.

This paper considers the problem of deploying a minimal number of (passive) observability points and (active) beacons in a network. We explicitly distinguish between observability points and beacons, which have *asymmetric roles*: in many cases, observability points and beacons have different implementations, require different resources, or come with different placement constraints or costs. For example, a beacon (an active *sender*) typically consumes more (networking) resources than an observability point (a passive *receiver*). On the other hand, deploying dedicated (passive) measurement nodes in the Internet core can be non-trivial and entail a significant investment compared to the deployment of light weight (active) measurement agents on the network edge (as e.g., in community-driven Internet measurement projects like DIMES [28]). Similarly, reading advertised routes from various BGP monitoring points may be significantly simpler than injecting new routes. To give an example in the context of enterprise networks, Ethernet root bridges naturally distribute distance information as part of the Spanning Tree Protocol (STP). Since root bridges can be configured by setting the bridge IDs, beacons are simple to deploy. On the other hand, observability points capturing and leveraging the STP packets for the detection of (and reaction to) link failures require changes in the network hardware and/or protocol headers. Such changes are typically infeasible in traditional, vertically integrated communication networks, and hence additional hardware needs to be deployed for observability points: for example, systems such as SHEAR [26] rely on OpenFlow switches to capture STP packets and render failover faster.

We are particularly interested in the formal study of the impact of the *routing model* on the efficiency of link failure detection. Interestingly, while the relevance of the routing model on path diversity [31] and hence tomographic power is intuitively clear, we lack a quantitative and formal evaluation. As we will show in this paper, the routing model has an impact already if we constrain ourselves to shortest paths with unit link weight only (routing *inside* most networks today is based on shortest paths, while *inter-network* routing usually is subject to complex policies). Indeed, modern computer networks often impose various constraints on the choice of shortest paths that can be selected for routing. For instance, in traditional communication networks, routing is typically *destination-based*: packets are forwarded according to the most specific destination prefix. Inter-domain routing is usually valley-free. Using Multiprotocol Label Switching (MPLS) or Software-Defined Network (SDNs) based Traffic Engineering, more general routing paths can be defined, e.g., routes which also depend on the source or which are (semi-)oblivious [20, 21]. In the presence of a load-balancer or Equal Cost Multi-Path (ECMP) control plane, multiple shortest paths may be monitored simultaneously between a beacon and an observability point. However, there also exist settings (e.g., in the Ethernet use

case discussed above [26]) where only the *distance* (but not the path) between beacons and observability points can be monitored. Accordingly, we will explore a spectrum of shortest path routing schemes in this paper, and present a taxonomy of routing protocols.

1.1 Our Contributions

This paper makes the following contributions:

1. *Asymmetric network tomography:* We introduce a natural network tomography model, which differentiates between probe senders and receivers (called beacons and observability points). While asymmetric tomography is a reality, we are not aware of any explicit and formal study.

2. *Impact of routing model:* We study the relationship between, and *quantify*, monitoring costs (the number of tomography nodes, i.e., beacons and observability points) and the routing model. In particular, we observe that knowledge of the network topology alone is insufficient to reason about the coverage of a given tomographic deployment. While in principle, this implies that a different optimal deployment needs to be computed for each routing model, we introduce a natural taxonomy for a canonical family of routing models with suitable algorithms to compute a deployment in this article.

3. *Empirical motivation:* We report on a small empirical study on Rocketfuel and Internet Topology Zoo networks, which confirms the impact of the routing model.

4. *Optimal and approximative algorithms:* We present optimal and near-optimal algorithms to deploy a minimal number of tomography nodes for link failure detection in different models for relevant *sparse* families of network topologies, namely cactus and outerplanar graphs (as we encountered them frequently in our empirical study). Moreover, we show that our results have implications on symmetric tomography as well.

5. *Computational hardness:* We show that the deployment problem is NP-hard in general. Moreover, we show that for some routing models, the problem is already computationally hard on simple and sparse network topologies such as cactus graphs.

6. *Attractive trade-offs:* We identify an interesting tradeoff between different routing schemes, in terms of monitoring power and the computational complexity. For example, one takeaway from our work is that it can sometimes be good to *artificially* restrict the routing model: while the path diversity does not suffer much from such a restriction, the computational complexity of deploying monitoring equipment can be reduced significantly (from NP-hard to polynomial-time solvable).

1.2 Organization

We provide intuition about asymmetric tomography and its limitations in Section 2 with some examples. Section 3 shows that the routing model critically

affects the complexity of tomography. We present optimal and approximative algorithms in Section 4. After reviewing related work in Section 5, we conclude our contribution in Section 6.

2 Asymmetry Matters

Before we highlight the difference between symmetric and asymmetric network tomography, we introduce some terminology. We model the network topology as a (connected) graph $G = (V, E)$, interconnecting nodes V with undirected links E ($|V| = n, |E| = m$). In order to be able to detect link failures, we deploy two types of tomography nodes, in the literature often also called *monitoring equipment* (shorthand ME), at different nodes of the network G: *beacons* and *observability points*. Formally, we describe the *deployment* as a mapping $\mu : V \to \{\mathrm{OP}, \mathrm{BC}, \mathrm{OP}+\mathrm{BC}, \emptyset\}$ which assigns to each node either an observability point, a beacon, both, or neither. In the following, we will refer to $\mathrm{BC}_\mu \subseteq V$ as the set of nodes selected in μ to function as *beacons*, and to $\mathrm{OP}_\mu \subseteq V$ as the set of nodes selected in μ to function as *observability points*. Let $\mu|_{G'}$ refer to the deployment restricted to the nodes of the subgraph G' of G. By slightly abusing notation, we will write $|\mu|$ to denote the total *deployment cost*, i.e., the total number of beacons and observability points. Moreover, we write μ^* to denote an optimal deployment, a deployment of minimum cost $|\mu^*|$.

We assume that each observability point can monitor links along *some* shortest paths, *to all beacons*. Which shortest paths are *routing-model consistent* and can be used for monitoring, depends on the routing model, discussed in the next section. In particular, we will show that simply knowing the network topology (the adjacency matrix) and the "routing rules" (e.g., symmetric, shortest path routing) is ambiguous and insufficient: we need additional knowledge on how packets are routed on the network topology.

We will assume that a link can be monitored if and only if it lies on at least one routing model-consistent shortest path between a beacon and an observability point. The set of all links monitored by a deployment μ is denoted by M_μ. Our objective hence will be to deploy beacons and observability points in such a manner that all links in G are monitored using the *minimum* number of monitoring equipment (sum of observability points and beacons) necessary, i.e., $M_\mu = E(G)$. We refer to this task as the *asymmetric tomographic node placement problem*. In other words, a deployment that monitors all links is called *valid*, if it also minimizes the cost, it is called *optimal*. For simplicity, we restrict ourselves to shortest paths with respect to unit link lengths; however our work can be extended for more general paths.

In this paper, we will show that the choice of shortest paths which can be used for tomography can influence the deployment cost significantly. In principle, there is always a trivial solution to the asymmetric tomographic node placement problem, on any graph and for any routing model: we can simply place both an OP and a BC at each node, i.e., $\forall v \in V : v \to \mathrm{OP} + \mathrm{BC}$. Since each node has a unique shortest path to each neighbor, namely the direct link, each link is monitored by a (OP,BC) pair. However, obviously, the resulting deployment can be far from minimal.

Note that intuitively, the asymmetric tomographic node placement problem can also be seen as some kind of graph covering problem [27]: Given a bipartite

graph of OP nodes and BC nodes, can we map the links of this bipartite graph on G (i.e., the routes between OP and BC nodes) such that all physical links are covered? We aim to minimize the number of nodes in the bipartite graph which cover the links (not requesting the graph to be complete bipartite gives more freedom, e.g., a valid embedding can assign both types of monitoring equipment to one vertex).

To provide intuition about asymmetric tomography and highlight some of its key features, in this section, we will abstract from the effects of the routing model and focus on simple networks where there is a *unique* shortest path between a given observability point and a given beacon. We will only later discuss how to refine the tomography problem if routing paths are not unique.

Graphs featuring unique shortest paths are called *geodetic* in graph theory. The canonical example are trees. However, the class also includes other graphs, for example ring graphs with an odd number of nodes (or equivalently, edges). To gain intuition we focus on these two graph classes in this section.

Theorem 1. *A tree T with ℓ leaves is monitored optimally with a total of ℓ* BC *and* OP *nodes (at least one each).*

Proof. We prove the lower and upper bounds in turn; finally we discuss the symmetry in the solutions.

Lower bound: We observe that if v is a node of degree 1 and w is its neighbor, a beacon or an observability point needs to be located at v to monitor link (v, w). It follows that for a graph G and a valid deployment μ, monitoring equipment must be available on each degree-one node: $\forall v \in V, \deg(v) = 1 \rightarrow \mu(v) \neq \emptyset$. There are ℓ such leaf nodes in a tree.

Upper bound: A single OP is sufficient to monitor the tree. Let $l_1, \ldots, l_\ell$ be an arbitrary ordering of the leaves of T. We define the following deployment: $\mu(l_1) = \text{OP}, \forall i \in [2, \ell], \mu(l_i) = \text{BC}, \forall v \in V(T) \setminus \{l_1, \ldots, l_\ell\} \mu(v) = \emptyset$. Let (i, j) be an edge of T, whose removal will divide the tree into two trees: subtree T_i which contains i and subtree T_j which contains j. W.l.o.g., assume $l_1 \in V(T_i)$, and observe that there must exist $k \in [2, \ell]$ s.t. $l_k \in V(T_j)$, in order to monitor edge (i, j).

Symmetry: The tomography problem exhibits symmetry in the tree: as long as there is at least one observability point located at a leaf, it will monitor all (unique) shortest paths to all leaves hosting beacons. I.e., optimal deployments on the tree are symmetric: OP and BC can be exchanged arbitrarily, as long as there is at least one of each kind left. □

We note that deploying monitoring functionality at tree leaves is particularly attractive in the context of modern datacenters which are often tree shaped (e.g, fat-trees, Clos, multi-rooted trees): the leaves are the servers, where functionality can be deployed easily and in software.

On the other hand, asymmetry quickly influences the placement of monitoring equipment, even in simple geodetic graphs.

Theorem 2. *Already in simple geodetic ring graphs (connecting nodes in a circular manner), the* OP *and* BC *roles are asymmetric: it is not always possible to switch the roles of an* OP *and a* BC *node.*

Proof. Consider a ring graph $G(V, E)$ with $V = \{v_0, \ldots, v_{n-1}\}$ and $E = \{(v_i, v_{(i+1) \mod n}) | i \in [0, n-1]\}$ of odd size n: it is easy to see that this graph is geodetic. Let $k =$

$(n-1)/4$ and assume k is integer. The following deployment is valid: $\mu(v_0) = \text{OP}, \mu(v_k) = \text{BC}, \mu(v_{2k}) = \text{OP}, \mu(v_{3k}) = \text{BC}$, as all links between each OP and its closest BC on both sides (at distance k or $k+1$) are monitored. If v_0 and v_k swap their roles, i.e., if we change $\mu(v_0)$ to BC and $\mu(v_k)$ to OP, then the links between v_0 and v_{3k} are not on the shortest path between any possible monitoring pair. Thus these links are not monitored, demonstrating the asymmetry and concluding the proof. □

We also note the importance of asymmetry in terms of costs. Clearly, if the costs of the different equipment types is similar, symmetric tomography yields a good approximation for asymmetric placements as well: we can simply replace each symmetric device with two asymmetric ones, which gives a 2-approximation for two types (asymmetric nodes need at least as many locations). However, in general, the approximation can be arbitrarily bad (namely linear in the number of nodes): For example, we only need one observability point in the tree, and can add beacons to the remaining leaves.

3 The Routing Model Matters

For many applications, the routing model plays a crucial role: knowledge of the topology alone is insufficient to reason about path diversity or the coverage of a given tomographic deployment. Indeed, in principle, for each concrete configuration of forwarding and routing rules, a different optimal deployment of network tomography equipment might exist. In this section, we will introduce a basic taxonomy of routing models. We will later study algorithms and monitoring deployments for these different models. Interestingly, we will also show that different routing models come with different computational complexities: there are routing models for which optimal tomographic deployments can be computed efficiently, while for others it is NP-hard. We will also show that the additional deployment cost for models that are easier to solve is sometimes very small. This introduces an interesting tradeoff.

3.1 Taxonomy and Hierarchy

We explore a spectrum of routing schemes, ranging from very low path diversity (the *intersection model*, short $\cap$, allows to monitor only links belonging to *all* shortest paths) to very high path diversity (the *union model*, short $\cup$, allows to monitor all links on all shortest paths), with two natural intermediate models called *confluent* ($>$) and *any* (exactly one, !). Let us first introduce some notation. Let G be a graph. Let M^R_{xy} be the edges on paths between x and y which are used to forward messages from x to y according to a given routing model R. Thus, M^R_{xy} is also the set of edges that are monitored if $\mu(x) = \text{BC}$, $\mu(y) = \text{OP}$, and $\mu(v) = \emptyset$ for all other nodes $v \in V \setminus \{x, y\}$. In a general form, the selection of paths is a function of x, y, the observability points and the beacons, and the routing model. We restrict ourselves to symmetric routing: $M^R_{xy} = M^R_{yx}$. Henceforth, if the routing model is clear from the context or irrelevant, we will sometimes not explicitly state it in the superscript.

We focus on shortest path routing: for any two nodes $x, y \in V(G)$, let $SP(x, y)$ be the set of shortest paths (abbrv. *sp*) between x and y. For $s \in$

$SP(x,y)$ we denote by $E(s)$ the edges of the shortest path s: $M_{xy} \subseteq E(s)$. More generally, we use $E(X)$ for the edges of the subgraph of G induced by the set X. We identify the following canonic routing strategies:

Definition 1 (Routing Models)**.** *We distinguish between four routing models which determine the shortest paths messages take and hence which links can be monitored:*

1. ***Union*** $\cup$***:*** *In the union model, all edges belonging to one or several shortest paths between x and y are monitored: $M^{\cup}_{xy} = \cup_{s \in SP(x,y)} E(s)$.*
2. ***Any (i.e., exactly one)*** $!$***:*** *In the any model, a single shortest path is monitored: $\exists s \in SP(x,y)$ s.t. $M^{!}_{xy} = E(s)$. This path is* arbitrary and given *for each source-destination pair, and forms part of the input to the problem.*
3. ***Confluent*** $>$***:*** *In the confluent model, the shortest path choice is determined by the message destination: $\exists s \in SP(x,y)$ s.t. $M^{>}_{xy} = E(s)$ and $\cup_{z \in V} M_{zy}$ is a tree, given as part of the input to the problem.*
4. ***Intersection*** $\cap$***:*** *In the intersection model, only links which belong to all shortest paths are monitored: $M^{\cap}_{xy} = \cap_{s \in SP(x,y)} E(s)$.*

At one end of the extreme, it is possible to observe *all* shortest paths between an observability point and all beacons: the $\cup$ model. For example, imagine a load-balancing network or a network with an *Equal Cost Multi-Path (ECMP)* [15] control plane (where, e.g., hash functions can be reverse engineered), or a network supporting source routing [18]. With MPLS Traffic Engineering or in a Software-Defined Network (SDN), monitoring packets can be forwarded along arbitrary paths [2, 29]: the exactly one $!$ model. Internet routing is typically *destination-based* (confluent $>$): packets are forwarded according to the most specific destination prefix. On the other end of the spectrum, there exist settings (e.g., our Ethernet example above [26]) where only the *distance* (i.e., the number of hops, but not the path) between beacons and observability point can be used to conclude if links have failed. This corresponds to the most restrictive model, intersection $\cap$: only if a link that is on all shortest paths fails, a link failure affects the distance between the monitoring equipment. Thus we can only derive that a link failure occurred from the hop-distance measurement for the links that belong to all shortest paths.

Note that these models form a hierarchy of increasingly flexible routing.

Theorem 3. *Let $\mu^*(G,R)$ denote the optimal deployment on network G according to routing model R. It holds that $|\mu^*(G,\cup)| \leq |\mu^*(G,!)| \leq |\mu^*(G,>)| \leq |\mu^*(G,\cap)|$.*

Proof. Let $\mu^*(G,R)$ be an optimal deployment of G with routing model R. Let R' be another routing model such that $\forall x,y \in V, M^{R}_{xy} \subseteq M^{R'}_{xy}$. Since $\mu(G,R)$ is a deployment, we have $\cup_{x \in \mathrm{OP}, y \in \mathrm{BC}} M^{R}_{xy} = E$. Thus, $\cup_{x \in \mathrm{OP}, y \in \mathrm{BC}} M^{R'}_{xy} = E$ holds: $\mu(G,R')$ is also a valid deployment on G. The theorem then follows from the fact that $M^{\cap}_{xy} \subseteq M^{>}_{xy} \subseteq M^{!}_{xy} \subseteq M^{\cup}_{xy}$. □

The routing model can hence be seen as a knob allowing to adapt the path diversity and hence monitoring power. On one end of the spectrum we have $\cup$:

a model where all paths are observable. The ! model allows to observe an arbitrary but single route, $>$ restricts paths to destinations, and $\cap$ models a scenario where shortest paths must be unique. The latter is relevant in settings where only distances between beacons and observability points are given, and hence, failures can only be observed when the distance between BC and OP changes.

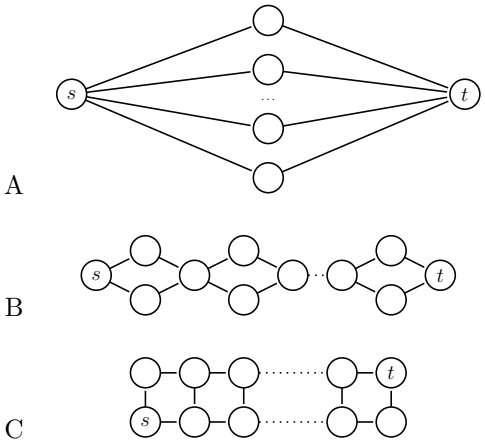

Figure 1: Three graphs illustrating the impact of the choice of routing policies on the cost of optimal monitoring. The *top graph* is a bipartite graph, the *middle graph* a one-connected cactus graph (a "sausage graph"), and *the bottom graph* an outerplanar graph (a "ladder graph"). In all these graphs the path diversity is high which also implies a high variance in the tomography costs for the different routing models. For example, $\cup$ requires two ME, one of each kind, placed on s and t, while more restrictive models like $\cap$ require up to a linear number of ME for the same graphs. See the text for more details.

To illustrate the impact of the routing model on monitoring efficiency, let us consider a few basic examples. In the best case, the entire network can be monitored with one BC and one OP (i.e., $|\mu^*| = 2$), and in the worst-case we need each kind of tomography node on each node. We will show that the different routing models can span the whole spectrum. Let us first consider bipartite graphs. Graph A depicted in Figure 1 is $B_{2,k}$: the complete bipartite graph connecting two nodes of the first node set with k nodes of the second node set. Between the two nodes of the first node set, k disjoint shortest paths exist. If the routing model is not $\cup$, each of the k shortest paths will have to be monitored separately, for a total monitoring cost of $k + 2$ ME. Graph B in the same figure is a chain of f planar faces. It is a one-connected graph belonging to the class of cactus graphs. Even though there exist 2^f different shortest paths between s and t, only two carefully selected such shortest paths allow to monitor all the edges between s and t. We therefore have $|\mu^*(G_B, \cup)| = 2$, $|\mu^*(G_B, !)| = 4$ (with one BC and one OP on s and on t, if the any-path is *choosable*, otherwise if it is *given* we have ≥ 4). For the other two routing models we need at least two ME per cycle, as will be proven in Corollary 1 later, thus $|\mu^*(G_B, >)| = |\mu^*(G_B, \cap)| = \Theta(f)$. Finally, graph C in the same figure is an

Graph class	$\cup$	$>$	!	$\cap$
Cactus	13.76%	1.49%	1.64%	1.98%
Outerplanar	15.96%	1.24%	2.43%	3.40%
General	17.06%	2.80%	3.16%	6.45%

Table 1: Percentage of nodes assigned OP for $\cup$ routing model, and percentage of additional OP compared with union when using the $>$, ! or $\cap$ routing model respectively. Results of a graph are counted towards the numbers for all graph classes it belongs to (and not just the most specific one).

outerplanar graph of f faces (see later for more details on outerplanar graphs), but these faces are connected through $f-1$ inner edges. It is 2-connected, and features $f+2$ different shortest paths between s and t. We have $|\mu^*(G_C, \cup)| = 2$, i.e., all paths can be monitored by placing a BC at s and an OP at t. For the model !, we can monitor at most $x \cdot y$ different paths with monitoring pairs composed out of x BC and y OP. Thus we need at least $\sqrt{f+2}$ monitoring equipment for this scenario $|\mu^*(G_C, !)| \geq \sqrt{f+2}$ deploying half of the ME as BC on the lower left nodes and the other half as OP on the upper right nodes. For the other routing models the amount of monitoring equipment we need is linear in the number of faces, $|\mu^*(G_C, >)| = |\mu^*(G_C, \cap)| = \Theta(f)$, see Theorem 11 proved later.

Theorem 4. *Depending on the model, the optimal deployment cost can vary by a factor $\Omega(n)$. This is worst possible.*

Proof. The outerplanar graph C in Figure 1 consists of $n/2-1$ faces with 4 edges each. As discussed above, the optimal monitoring cost is $|\mu^*(G_C, \cup)| = 2$, $|\mu^*(G_C, !)| \geq \sqrt{n/2+1}$, $|\mu^*(G_C, >)| = |\mu^*(G_C, \cap)| \geq n/2-1$. □

3.2 Empirical Results

We have seen that the routing model can in principle have a large impact on the tomography cost. In order to study whether this is only the case for contrived examples and when computing an optimal deployment, we conducted a small empirical study using a simple greedy algorithm (see Algorithm 3) and considering two sets of real-world topologies: the Internet Topology Zoo [19] and the Rocketfuel graphs [30].

For our empirical evaluation, we implemented the routing restrictions for the any ! model and the confluent $>$ model as follows. For the any ! model, we select the first shortest path between two nodes computed by the Python NetworkX library. For the confluent $>$ model, we use the breadth-first tree computed using the Python NetworkX library for each observability point o, to ensure that the monitored paths from all beacons are confluent.

As a first and independent observation, we find that the studied graphs are often sparse. In fact, almost a third (32%) of the parseable graphs belong to the family of *cactus graphs*: A cactus graph (sometimes also called a cactus *tree*) is a connected graph in which any two simple cycles have at most one node in common. In other words, in a cactus graph, any link belongs to at most one cycle, and hence, a cactus graph can be decomposed such that every link is

either a part of a single cycle or a part of a single tree (the line is treated as an atrophied tree). By definition, different cycles and trees may share at most one node. We call the articulation points connecting cycles and/or trees *nexus nodes*. A unicyclic graph is a cactus with exactly one cycle.

Besides cactus graphs, roughly half of the graphs (49%) are *outerplanar*: an outerplanar graph is a planar graph which can be drawn in such a manner that each node touches the outer-face. Outerplanar graphs are hence a generalization of cactus graphs.

When analyzing the the deployment cost, we find that both the graph class and the routing model indeed influence the number of required monitoring equipment. Table 1 summarizes our results. Using the $\cup$ model as the baseline, the table shows for the different routing models what percentage of nodes are OP according to the computed deployment for this model, and what the additional percentages of monitoring nodes are for the other models. Not surprisingly, we can see that the number of equipment assigned grows for more general graph classes and for more restrictive models from around 13% to around 23%. The percentage of BC varies similarly at around 70%.

We also find that between some models the difference may not be large (below 3.5% on average for most models and classes). This could be exploited, as some routing models are cheaper to implement than others (e.g., due to number of routing rules needed at each node).

3.3 Complexity

In addition to the differences mentioned above, the computational complexity of the underlying algorithmic tomography problem can vary as well. For example, the !-routing model is NP-hard on cactus graphs, however, as we will see, cactus graphs can be solved efficiently for other routing models. Moreover, a destination-based routing model may require less forwarding rules than oblivious routing models which also depend on the traffic source.

Theorem 5. *The asymmetric tomographic node placement problem on cactus graphs is NP-hard under the any* !*-routing model.*

Proof. We provide a reduction from the NP-hard problem Set Cover [13]. The input of the set cover problem is a set of n elements and m subsets $S_1, \ldots, S_m$ containing some of the elements, and an integer k. The output should be *true* iff there is a selection of k of the subsets such that their union includes all n elements.

Given a set cover problem instance, we demonstrate how to construct a cactus graph $G(V, E)$ and a shortest path routing on it such that a solution of the asymmetric tomographic node placement problem on G can be used to derive a solution to the set cover problem.

We take the "sausage" graph depicted as graph B in Figure 1 with n even length cycles and add a "line" of length $m+1$ nodes connected to their left and right neighbor on the right end. Each of the n cycles stands for an element. A shortest routing path from node v_l at distance l from the right end to the left-most node encodes the subset S_l: for each element that is in the set, the path on the top of its corresponding cycle is taken, whereas the lower path is used otherwise. For all other shortest path pairs the lower path is taken whenever feasible.

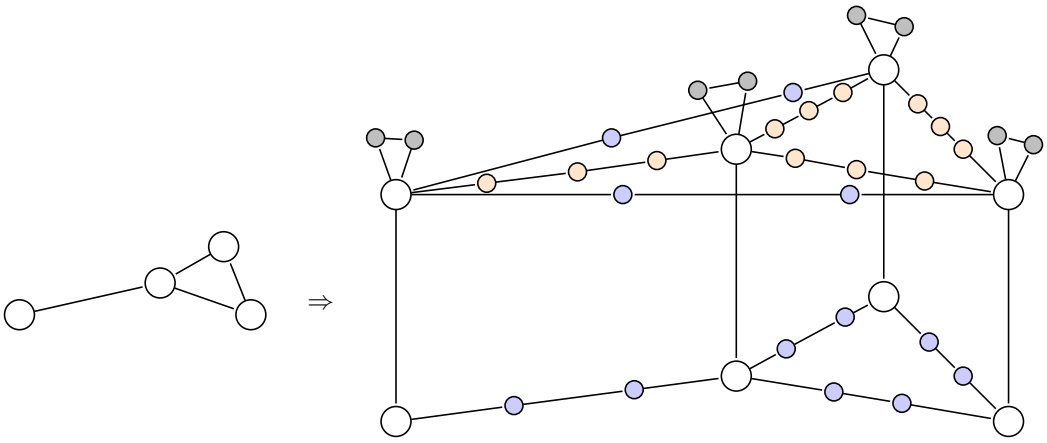

Figure 2: Illustration of construction of $G'(V', E')$ (right) given a graph $G(V, E)$ (left). The nodes of V and W are shown as white circles, nodes of W' and W' are gray, nodes of three-hop chains are blue and nodes of four-hop chains are orange. In the proof we show that a k-node cover of G is equivalent to a $(2n+k)$ deployment in G'.

Without loss of generality we can put an OP on the left-most node and a BC on the right. This monitors all edges of the lower from left to right. To monitor the top edges, either a ME has to be placed on the corresponding top node, or on one of the nodes at distance 1 to l from the right. It is easy to see that iff there is a valid deployment with k ME, there is also a solution to the set cover problem: a valid deplyoment with a BC on node v_l corresponds to a set cover solution with set S_l. For an ME on the cycle corresponding to element i pick any set S_l which contains i. □

While we will present fast and optimal algorithms for some sparse graphs later in this paper, we note that the problem is NP-hard on general graphs for all routing models.

Theorem 6. *The asymmetric tomographic node placement problem is NP-hard, under all our routing models* $\cup$, !, $>$, *and* $\cap$.

Proof. We consider the decision problem, where we are given a deployment cost threshold k. We provide a reduction from the NP-hard Node Cover problem [13]. The node cover problem asks whether for a given graph $G(V, E)$ and a threshold k, there exists a subset $S \subseteq V(G)$ such that $\forall (v_i, v_j) \in E(G),\ \{u, v\} \cap S \neq \emptyset$ and $|S| \leq k$.

Given such a node cover problem instance, we show how to efficiently build a graph $G'(V', E')$ such that a solution of the asymmetric tomographic node placement problem on G' can be used to derive a solution to the node cover problem. In order to prove the hardness for all routing models $\cup$, !, $\leq$, and $\cap$ simultaneously, we build a problem instance where all breadth first shortest path trees are either unique, or where the branches needed to monitor additional edges are unique.

Given $G(V, E)$, construct $G'(V', E')$ as follows. Add the node sets W, consisting of $3n$ vertices, w_i, w_i' and w_i'' for each $v_i \in V$. Add edges between the

vertices w_i, w'_i, w''_i so each triple forms a triangle. Build a clique among the nodes w_i. Connect w_i with v_i. Then replace each edge $(v_i, v_j) \in E$ with a 3-hop chain, replace each edge (w_i, w_j) where $(v_i, v_j) \in E$ with a 4-hop chain, and replace each edge (w_i, w_j) where $(v_i, v_j) \notin E$ with a 3-hop chain. An example for a small graph $G(V, E)$ is depicted in Figure 2.

We will now show that in this constructed graph G', a deployment with $2n + k$ monitoring equipment exists if and only if there is a node cover of G with k nodes.

To this end, note that all triangles require at least one beacon and one observability point, otherwise the edge between w'_i and w''_i is not monitored: it is on the shortest path between any other pairs of nodes. W.l.o.g., let us put equipment of one type (say a beacon) on w'_i and the other one (the observability point) on w''_i for all i. This deployment monitors all edges on paths among nodes in W. Adding further monitoring equipment on nodes of W, or on nodes on paths between nodes in W, does not increase the number of edges monitored. Thus nodes in V or on paths between V need to be equipped.

Observe that there is a unique shortest path from v_i to w_j of length four if $(v_i, v_j) \in E$, going through v_j. If $(v_i, v_l) \notin E$ then there is a unique shortest path from v_i to w_l of length four, not passing any other node $v_l \in V$. Thus, assigning monitoring equipment on a node v_i helps to monitor paths corresponding to edges to neighbors of v_i in G and the edges from v_i and its neighbors to W. Paths corresponding to two-hop neighbors in G cannot be monitored by this deployment. Next, let us consider, a node u on a three-hop path between v_i and v_j where $(v_i, v_j) \in E$. Without loss of generality, we can assume that u is at distance one from v_i. Assigning monitoring equipment to u lets us monitor exactly the same edges as assigning it to v_i, because the union of the shortest paths $SP(u, w_i)$, $SP(u, w_j)$ and $SP(u, w_l)$ where $(v_i, v_l) \notin E$ use the same edges as the union of $SP(v, w_i)$, $SP(v, w_j)$ and $SP(v, w_l)$. Hence u and v_i are equivalent with regards to monitoring; edges of E' which only appear on paths to nodes v_w, where $(v_i, v_w) \notin E$, are not monitored.

As a consequence, choosing a node cover set of $G(V, E)$ and assigning equipment to it, guarantees that all edges are monitored. On the other hand, any set of nodes that does not cover all edges, also fails to monitor all edges of E'. Given a valid deployment of $G'(V', E')$ with $2n + k$ nodes the assigned monitoring equipment can be used to construct a node cover for $G(V, E)$ with k nodes. Let S be the set of nodes corresponding to V and the nodes on the shortest paths in G' between V with a monitoring equipment deployment. For each node x in S, add the node $v \in V$ closest to x to the node cover. □

4 Optimal and Approximative Algorithms

Given our insights into the properties of asymmetric tomography and the dependency on the routing model, and motivated by the sparse structure of the topologies collected in the previous section, we now devise deployment algorithms for two sparse graph families: *cactus graphs* and *outerplanar graphs*.

ALGORITHM 1: EquipCycle(C,μ)

Let P be the subgraph of C where edges monitored by μ are removed: $P = (V(C), E(C) \setminus \cup_{o \in \mathrm{OP}_\mu, b \in \mathrm{BC}_\mu} M_{ob})$
Let μ' be an empty assigment
while *unmonitored edges exist,* $E(P) \neq \emptyset$ **do**
    $(v, me) = \arg\max_{x \in V(P), m \in \mathrm{ME}} |\{\text{newly monitored links in } P\}|$
    $\mu'(v) = me$
    $P = P \setminus \{\text{all new edges monitored by } me \text{ on } v\}$.
return μ'

4.1 Cactus Graphs

This section presents polynomial-time optimal deployment algorithms for cactus graphs and the models $\cup, >, \cap$.

Lemma 1. *Let μ be a deployment with at least one* OP *and one* BC *on a cycle C. If there are edges which are not monitored by μ, then they form a (single) connected path.*

Proof. Let $S = E(C) \setminus M_\mu$, where M_μ is the set of all edges monitored by the deployment μ. For the sake of contradiction, assume that the set S of not (yet) monitored edges is not a path. Then there are at least two connected components which are not monitored. Since $S \subseteq C$ and C is a cycle, at least two pairs of monitoring equipment partition C in two parts. Let (O_1, B_1) and (O_2, B_2) be these pairs. In this case, paths from O_1 to B_2 and O_2 to B_1 are monitored and thus there are no edges which are not monitored. □

From Lemma 1 it follows that two MEs are required for $\cup$ in even-length cycles. In all other models, four equipments are required: with three equipments, there always exists an edge between two nodes of the same equipment which is not monitored. We have the following corollary:

Corollary 1. *On a cycle, two or four* ME*s are required for even and odd cycle lengths respectively. For the models $>$ and $\cap$, four* ME*s are necessary, for any cycle length.*

Using the above, we can devise an algorithm that adds equipment in a greedy fashion for the routing models $\cup, >$ and $\cap$. Given a deployment μ for a cycle C, with an unmonitored path P, an additional equipment m is assigned to v such that the maximum number of previously unmonitored links is now monitored, i.e., $(v, me) = \arg\max_{x \in V(P), m \in \mathrm{ME}} |\{\text{newly monitored links in } P\}|$, breaking ties arbitrarily.

Theorem 7. *Let μ be a possibly empty deployment of* ME *on a cycle C. Algorithm 1 determines a valid deployment for all models and an optimal completion of the deployment in a greedy fashion for the models $\cup, >, \cap$.*

The proof is technical and postponed to the Appendix (Section 7.1).

As a next step we show that for graphs consisting of two subgraphs "merged" at a common nexus node, their deployments can be computed separately under certain conditions. This is very useful, as it enables a "divide and conquer" approach in loosely connected graphs.

Definition 2 (Graph merging)**.** *Let G_1, G_2 be two arbitrary graphs, and let $v \in V(G_1), v' \in V(G_2)$. We define the* graph merging *operation $G_1vv'G_2 = G$ as the contraction of vertices v and v' into a single node, thus connecting G_1 and G_2. For deployments $\mu|_{G_1}$ and $\mu|_{G_2}$, we write $\mu = \mu|_{G_1} + \mu|_{G_2}$ to refer to their composition.*

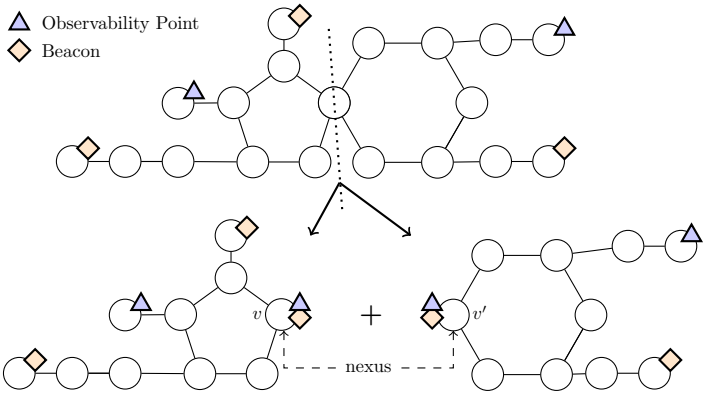

Figure 3: Illustration of the Partial Deployment approach (Lemma 2): a graph G (above) is decomposed into two subgraphs G_1 and G_2 which are equipped independently. In these intermediary steps, nexus nodes are virtually equipped with the monitoring equipment deployed in the subgraph they connect to. This approach preserves optimality (see Lemma 3).

If we "move" all ME from one subgraph G_2 to the nexus node, we can determine a valid deployment of the other subgraph G_1 without considering G_2's structure. Figure 3 illustrates this approach: the nexus node connecting the two cycles of G (above) is represented in *each* subgraph G_1 (bottom left) and G_2 (bottom right). Nexus nodes are then virtually equipped with monitoring equipment contained in the subgraph they connect; here as both G_1 and G_2 contain both equipment types, v and v' are each equipped with a beacon and an observability point. To compute a deployment for one of the subgraphs when the other subgraph has been equipped already, the virtual deployment on the nexus node can be taken into account. The intuition for the correctness of this approach relies on the fact that any shortest path between nodes of G_1 and nodes of G_2 will necessarily cross the nexus nodes. From G_1's perspective the precise location of monitoring equipment in G_2 does not matter –only knowledge about the existence is important. This allows the deployment of both subgraphs to be optimized independently.

Lemma 2 (Partial Deployment)**.** *Let G_1, G_2 be two arbitrary graphs, and $v \in V(G_1), v' \in V(G_2)$. Let μ be a deployment on $G = G_1vv'G_2$. Let μ_r be the deployment that assigns all* ME*-types deployed on G_2 to node v: $\mu_r(v) = \cup_{x \in V(G_2)} \mu(x)$. Let μ_1 be an deployment on G_1. It holds that $E(G_1) \cap M_{\mu_1+\mu_r} = E(G_1) \cap M_{\mu_1+\mu|_{G_2}}$: all edges monitored in G_1 by μ_1 are still monitored when merging with G_2 using $\mu|_{G_2}$ under $\cup, >, \cap$.*

Proof. Let $\mu_C = \mu_1 + \mu|_{G_2}$. Assume there are edges in G_1 which are monitored by $\mu_1 + \mu_r$ but not by $\mu_1 + \mu|_{G_2}$, i.e., $\exists e \in E(G_1) \cap M_{\mu_1+\mu_r}$ s.t. $e \notin E(G_1) \cap$

$M_{\mu_1+\mu|_{G_2}}$. Since e is monitored by $\mu_1+\mu_r$, $\exists x \in \mathrm{BC}_{\mu_1+\mu_r}, y \in \mathrm{OP}_{\mu_1+\mu_r},\ s.t.\ e \in M_{xy}$. If $x \in \mathrm{BC}_{\mu_1} \wedge y \in \mathrm{OP}_{\mu_1}$, then necessarily $x \in \mathrm{BC}_{\mu_c} \wedge y \in \mathrm{OP}_{\mu_c}$. Therefore $x \in \mathrm{BC}_{\mu_r} \vee y \in \mathrm{OP}_{\mu_r}$. Assume w.l.o.g., $y \in \mathrm{OP}_{\mu_r}$, then necessarily $y = v$, and $e \in M_{xv}$.

However since $\mu_r = \cup_{v \in V(G_2)} \mu(v)$, $\exists y' \in V(G_2)\ s.t.\ y' \in \mathrm{OP}_{\mu_2}$. Because all shortest paths from G_1 to G_2 go through v, we have that v is an endpoint of an edge in M_{xv} and $M_{xy'}$, and due to definition x is an endpoint of any edge in $M_{x\cdot}$, $e \in M_{xy'}$ for the routing models $>, \cap$: e is monitored. Of course, as $e \in SP(x, v)$, it is also monitored in the $\cup$ routing model.

Now assume the opposite situation $\exists e \notin E(G_1) \cap M_{\mu_1+\mu_r} \wedge e \in E(G_1) \cap M_{\mu_1+\mu|_{G_2}}$. Edge e is necessarily on a monitored path from a ME in G_1 on u to a ME in G_2 on w. This path must go through v, which is equipped with all the monitoring equipment of G_2 by μ_r. Therefore e must be in $M^{\cup}_{\mu_1+\mu_r}$ as all shortest paths in $M^{\cup}_{uv}$ are contained in $M^{\cup}_{uw}$. Analogously it holds that $M^{\cap}_{uv} \subseteq M^{\cap}_{uw}$. For the confluent model with symmetric routes, it holds that $M^{>}_{xy} = M^{>}_{yx}$ and hence $M^{>}_{uv} \subseteq M^{>}_{uw}$. Since $u \in V_1, w \in V_2, v \in V_1 \cap V_2$, we deduce $M_{uw} \cap E(G_1) = M_{uv}$. The fact that $e \in E(G_1)$ and $e \in M_{uw}$, implies that $e \in M_{uv}$. Since any ME in G_2 is deployed on v, $\mu_C(w) \subseteq \mu_r(v)$, and thus $e \in M_{\mu_1+\mu_r}$. □

Such a partial deployment does even preserve optimality: an optimal deployment for G_1 can be computed separately from G_2 by assuming equipment on the nexus node connecting them.

Lemma 3 (Partial Deployment Optimality)**.** *Consider two arbitrary graphs G_1, G_2 and $v \in V(G_1), v' \in V(G_2)$. Assume μ^* is an optimal deployment on $G = G_1 v v' G_2$ and μ_r assigns all ME of $\mu|_{G_2}$ on v as in Lemma 2. If $\mu_1 + \mu_r$ is an optimal deployment on G_1, it must hold that $\mu_1 + \mu|_{G_2}$ is an optimal deployment of G under $\cup, >, \cap$.*

Proof. Let $\mu_C = \mu_1 + \mu|_{G_2}$. Thanks to Lemma 2, we know that μ_C is a valid deployment of G. Assume that μ_C is not optimal. Since μ is optimal we have $|\mu_C| > |\mu|$. Both μ and μ_C are identical on the G_2 part: $|\mu| = |\mu|_{G_1}| + |\mu|_{G_2}| < |\mu_C| = |\mu_1| + |\mu|_{G_2}|$. Thus a difference in the number of equipment must manifest in G_1.

Thus necessarily $|\mu|_{G_1}| < |\mu_1|$. Since μ is valid, it monitors all edges of G, and in particular all edges of G_1, we thus conclude $M_{\mu|_{G_1}+\mu_r} = E(G_1)$. Thus $\mu|_{G_1} + \mu_r$ defines a valid deployment on G_1 of size $|\mu|_{G_1}| + |\mu_r| < |\mu_1| + |\mu_r|$: this contradicts the definition of $\mu_1 + \mu_r$ as optimal on G_1. □

We can apply Lemma 3 recursively to compute an optimal deployment for cacti. This is the approach followed by Algorithm 2: it first assigns monitoring equipment to leaves, and then processes the cycles in a specific order. Processing cycles in a bottom up approach from the leaves gradually towards the center allows us to take decisions on each individual cycle separately while only one of its nexus nodes is not yet equipped (π_u in the algorithm), since other nexus nodes either connect to leaf nodes or to lower cycles that are already equipped.

Lemma 4. *Algorithm 2 produces a valid deployment μ for a cactus graph G under the routing models $\cup, !, >$ and $\cap$.*

ALGORITHM 2: Optimal Deployment for Cactus $G(V, E)$

Let T be the contracted tree of G (for each cycle with k nexus nodes, the nexus nodes are kept and the cycle is replaced with a node connected to the cycle's nexus nodes)
$C \subset V(T)$: set of vertices of T corresponding to cycles in G
L_0: set of leaves of T. Inductively define L_i as the leaves of the tree induced by $V(T) \setminus (\cup_{j<i} L_j)$, for $i > 0$
$h = \arg\max_{j>0}\{L_j \neq \emptyset\}$

/* Process leaves */
if *G contains more than one cycle, $|C| > 1$* **then**
    Let r be a node located between two cycles of G
else /*G has a single cycle*/
    Let r be a nexus with the most leave nodes.
for *each leaf node v in depth-first order from r, respecting the clockwise order of a given embedding* **do**
    Alternatingly assign $\mu(v) = \text{OP}$ or $\mu(v) = \text{BC}$

/* Process cycles */
for $i = 0..h$ **do**
    for *each cycle C_j on layer L_i* **do**
        Let μ' be a temporary virtual deployment on C_j, $\mu'(C_j) := \emptyset$
        for *each nexus π of C_j* **do**
            Let G_π be the subgraph connected to π
            if *G_π contains a cycle on L_{i+1}* **then**
                $\mu'(\pi) = \text{OP} + \text{BC}$
            else
                π is virtually equipped with all ME in G_π,
                $\mu'(\pi) = \cup_{v \in V(G_\pi)} \mu(v)$
        $\mu = \mu \cup \mathit{EquipCycle}(C_j, \mu')$
return μ

Proof. Edges of G are either part of cycles or part of T. Since by construction a valid deployment on T is realized, all non-cycle edges of G are monitored. Since every cycle belongs to a layer L_i, it will be processed by *EquipCycle*. Due to Lemma 2, edges that are monitored as a result of *EquipCycle* will still be monitored in G, provided the nexus nodes are correctly represented for the models $\cap, \cup, >$.

Thanks to the bottom-up approach, knowing what a nexus will monitor is simple: in every cycle at most one nexus (say π_u) connects to the upper layer. Thus for all the other layers below, the ME deployment is already known. Let G_u be the subgraph connected to π_u. If G_u contains a cycle, necessarily G_u will contain OP + BC for all routing models different from $\cup$, and π_u will receive a correct deployment in Line 17. Otherwise, G_u does not contain any cycle, therefore it only contains leaf nodes that are already assigned: π_u will be correctly initialized in Line 19. To compute a valid deployment for the ! model, we can execute the algorithm for the $\cap$ model and apply Theorem 3 to derive the correctness. □

In case of $>$ and $\cap$, the computed deployments are also optimal.

Theorem 8. *Algorithm 2 is optimal for models $>$ and $\cap$.*

The proof is technical and appears in the Appendix (Section 7.2).

We have so far discussed asymmetric placement algorithms for scenarios where different equipment types are of the same cost. In order to account for asymmetric costs, we can postprocess the deployment μ computed by Algorithm 2, and swap more expensive with cheaper types, while preserving the monitoring properties. First, all ME in the deployment μ are replaced by the cheaper equipment, BC. Subsequently, each cycle is processed separately to equip it optimally with at most two OP. To this end, all nexus nodes are virtually equipped with BC + OP if there is another cycle in the connected subgraph, and with BC otherwise. Given this equipment, for a cycle with k nodes, there are at most $\binom{k}{2}$ options to consider when deciding where to replace a BC with an OP. Thanks to Lemma 3, this leads to an optimal deployment for the models $\cap$ and ¿.

4.2 Outerplanar Graphs

Our optimal algorithm for cactus graphs raises the question whether good polynomial-time solutions also exist for more general graph classes. A natural generalization of cactus graphs are *outerplanar graphs*: a graph is called outerplanar iff it can be drawn in the plane without crossings, in such a way that all of the vertices belong to the (unbounded) face of the drawing. In other words, no node is totally surrounded by edges. Clearly, cactus graphs are outerplanar: when cycles are merged into a single node, a cactus graph becomes a tree. Recall that in the Internet Topology Zoo [19], nearly half of all topologies are outerplanar.

We now show that there exists a simple greedy approximation algorithm for outerplanar graphs. More precisely, we devise an algorithm that produces a valid deployment for arbitrary graphs and then derive its approximation ratio for outerplanar graphs. The algorithm picks an arbitrary node r and builds a breadth-first tree B from there, such that the routing model $M \in \{\cup, !, >\}$ is

adhered to, i.e., messages are forwarded according to B (and potentially other links in addition). The root r is assigned an OP and all leaves of B are assigned a BC. This guarantees that all edges of B are monitored. For the remaining edges, i.e., the edges which close cycles put one ME on one of the incident nodes and one ME of the other type on the other incident node if necessary. A description in pseudocode is provided in Algorithm 3.

ALGORITHM 3: Valid deployment for arbitrary graphs

```
1 Construct Breadth-First Tree (BFT) B from arbitrary node r according to
    model M
2 μ(r) = OP
3 for each leaf node v of B do
4     μ(v) = BC
5 while ∃ unmonitored edge e = (u, v) ∈ E do
6     μ(u) = μ(u) + BC, μ(v) = μ(v) + OP
```

Theorem 9. *Algorithm 3 computes a deployment for any given graph with at most* $\min(2n, m+1)$ *equipment cost for routing models* $\cup, !$ *and* $>$.

Proof. Since the breadth-first tree B has been produced in line with the routing model, all its edges are on legal shortest paths between ME of different kinds after Line 3. Subsequently, executing the while loop for each edge ensures that all remaining edges are monitored as well. Equipping the tree B uses at most $n-1$ BC and one OP. On the remaining $k = m - n + 1$ edges at most kBC and kOP nodes are deployed. Thus the total cost of the deployment is $(n-1)\text{BC} + \min(n, m-n+2)\text{OP} = \min(2n, m+1)\text{ME}$. □

Despite its simplicity, Algorithm 3 provides good results. To show this, we evaluate its cost and compare it to the cost of an optimal deployment μ^*. For confluent routing, it is easy to show that this algorithm computes a 2-approximation for cactus graphs.

Theorem 10. *Algorithm 3 computes a 2-approximation for cactus graphs for the models* $\cup$ *and* $>$.

Proof. Note that $\mu^*(G, >)$ assigns at least one ME to each leaf and to each cycle with one or two nexus nodes. Cycles with more than two nexus nodes result in either another leaf or another leaf cycle with one nexus node. On the other hand, μ constructed by Algorithm 3 assigns at most two ME to each cycle, thus $|\mu|/|\mu^*| \leq 2$. □

Let us now turn to outerplanar graphs. They provide interesting insights into the impact of routing on monitoring cost. Evaluating the cost of an optimal deployment μ^* for outerplanar graphs on the other hand is challenging, and we resort to lower-bound the cost of an optimal deployment: we decompose an outerplanar graph into a set of faces, and show that peripheral faces need at least one monitoring equipment (Lemma 5), while inner faces require a different counting strategy, as presented in Lemma 6.

Technically, our proof approach heavily relies on the tree structure the outerplanar faces produce. More precisely, we make use of the concept of *weak planar duality*: The *dual graph* of a planar graph G is a graph that has a node for each

face of G; the dual graph has an edge whenever two faces of G are separated from each other by an edge. Thus, each edge e of G has a corresponding dual edge, the edge that connects the two faces on either side of e. The *weak dual* of a planar graph is the subgraph of the dual graph whose vertices correspond to the bounded faces of the primal graph. A planar graph is outerplanar if and only if its weak dual is a forest. An example graph and its weak planar dual is depicted in Figure 4.

We begin by proving that a valid monitoring deployment places at least one ME on faces of degree one in the weak dual forest.

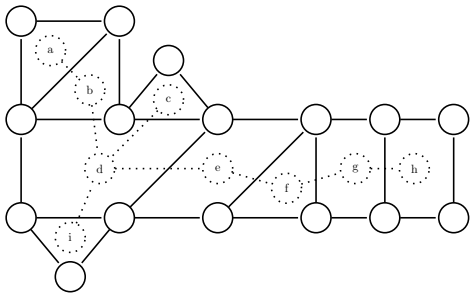

Figure 4: Outerplanar graph G with its weak dual tree T depicted using dotted vertices, representing the faces of the graph G, and using dotted edges between neighboring faces. In this example, the faces a, c, h, i are of degree 1 in the tree T, while b, e, f, g are "chain" faces of degree 2. Face d is of degree 4.

Lemma 5. *Let G be an outerplanar graph, where the corresponding weak planar dual is a tree T, $\deg_T(F) \geq 1$ for all faces F . Let μ be an optimal deployment for $\cup, >$, or $\cap$. Let F be a leaf face, $\deg(F) = 1$. Then necessarily $\exists v \in V(F)$ s.t. $\mu(v) \neq \emptyset$.*

Proof. The proof is by contradiction: assume that $\forall v \in V(F), \mu(v) = \emptyset$ and that F is nevertheless monitored. Let (l, r) be the two nexus nodes connecting F to the rest of G. Since G is outerplanar, we have $(l, r) \in E(G)$.

Let $e \in E(F) \setminus \{(l, r)\}$ be an edge of the face that is not in G. Since F is monitored (i.e., all links making up the boundary of the face are monitored), e must be on a shortest path between two nodes with ME: $\exists (x, y) \in V(G \backslash F)$ s.t. $\mu(x) = \text{BC}, \mu(y) = \text{OP}$ and $e \in M_{xy}$, thus $e \in SP(x, y)$. W.l.o.g., we can decompose the shortest path between x and y as follows: $SP(x, y) = SP(x, l) + P_a + e + P_b + SP(r, y)$ with P_a and P_b in F. Let $P' = SP(x, l) + (l, r) + SP(r, y)$: since $|P_a| + |P_b| > 1$, we have $|P'| < |SP(x, y)|$: P' is a path connecting y and y that is shorter than the shortest path, a contradiction □

At this point, an important observation is that no single shortest path between a BC and an OP can monitor a complete face alone, since a face by definition creates a loop that cannot exist in shortest paths. A second similar observation is the following: if a face is monitored by two paths, then these paths necessarily intersect. Both observations can be translated into constraints on the minimum number of equipment on the parts of G that form "chains" in the dual tree T. More formally, let C_x be a chain of x faces in T: there exist

two trees L, R s.t. $T = LC_xR$ and each face of C_x is of degree 2. In Figure 4, the faces e, f, g form a chain of length 3, as an example.

Lemma 6. *Let G be an outerplanar graph, where the corresponding weak planar dual is a tree T, $\deg_T(F) \geq 1$ for all faces F. Let C_x be a chain of x faces in T: $\exists L, R$ two trees s.t. $T = LC_xR$ and each face of C_x is of degree 2. For an optimal deployment μ it holds that $|\mu^*| \geq 2$ for the routing model $\cup$, $|\mu^*| \geq \sqrt{x}$ for the model ! and $|\mu^*| > (x-3)/2$ for the model $>$.*

Proof. Let $(c_l, c'_l) = E(L) \cap E(C_x)$ and $(c_r, c'_r) = E(R) \cap E(C_x)$ the edges of the faces connecting C_x with L and R respectively (if L or R must contain at least one additional node each, otherwise C_x could not contain faces of degree two only). We can partition $E(C_x)$ in three groups. Let F_j be the j^{th} face of C_x from L's side and $I = \{i_1, \ldots, i_{x-1}\} = \{e \text{ s.t. } j < x \wedge E(F_j) \cap E(F_{j+1}) = \{e\}\}$ be the set of inner edges shared by more than one face. Observe that $C_x - (I \cup \{(c_l, c'_l), (c_r, c'_r)\})$ contains two chains (one being possibly empty). Let U and D refer to one of the chains each, and let U_i and D_i be the corresponding subchains for each face; in case a face F_i does not contain any edge on a side, we slightly abuse the notation and write $U_i = \{(v, v)\}$ or $D_i = \{(v, v)\}$.

Let $l \in L$ and $r \in R$ be two arbitrary nodes of the left and right subtrees. Observe for a given shortest path $p \in SP(l, r)$ between these nodes that $U_i \cap p \neq \emptyset \Leftrightarrow D_i \cap p = \emptyset$ and vice-versa. It is thus possible to decompose p as a set $K = \{k_1, k_2, \ldots, \}$ of $|K| < x$ side switches from U to D or vice-versa, taking place using edges i_{k_1}, i_{k_2} etc. Hence s.t. $p \cap C_x = [U_1, \ldots, U_{k_1}, i_{k_1}, D_{k_1+1}, \ldots, D_{k_2}, i_{k_2}, U_{k_2+1}, \ldots]$ *or* its "complement" path $[D_1, \ldots, D_{k_1}, i_{k_1}, U_{k_1+1}, \ldots U_{k_2}, i_{k_2}, D_{k_2+1}, \ldots]$. This allows us to express the impact of the routing model: any two shortest paths between L and R monitoring edges of I must intersect. If this is not restricted by the routing model (under the $\cup$ routing model all shortest paths are monitored), there are configurations where only one pair of monitoring equipment is necessary to monitor all edges of C_x. If the routing path intersections are constrained as in the ! model, a distinct (l, r) pair is needed for each edge of I. In the best case, using m monitoring equipment, it is possible to create m^2 distinct shortest paths, thus $|\mu^*| \geq \sqrt{x}$. Observe that in the confluent $>$ model, all monitoring paths between l and r use either $M^{>}_{c_l,c_r}$, $M^{>}_{c'_l,c_r}$, $M^{>}_{c_l,c'_r}$, or $M^{>}_{c'_l,c'_r}$. Thus at most three faces can be fully monitored by ME exclusively in L or R. For each pair of other faces, at least one ME on C_x needs to be deployed. Thus the number of MEs on C_x is at least $(x-3)/2$, thus $|\mu^*(G, >)| \geq (x-3)/2$. □

With both the leaf face and the chained face cases bounded, we can now establish a lower bound:

Theorem 11. *Given an outerplanar graph G and its weak planar dual graph T, $\deg(F) \geq 1 \forall F \in T$, let μ^* be an optimal deployment and $d_i = |\{F \in T, deg(F) = i\}|$. For the any routing model !, $|\mu^*(G, !)| \geq \max(\sqrt{d_2}, (|T| - d_2)/2)$. If routing is confluent $>$, then $|\mu^*| \geq \max(2, (|T| - 3)/2)$.*

Proof. Observe that $d_1 = 2 + \sum_{i>2}(i-2) \cdot d_i \geq \sum_{i>2} d_i$. Since $|T| = d_1 + d_2 + \sum_{i>2} d_i$, we have $d_1 \geq (|T| - d_2)/2$. The theorem follows since $|\mu^*| \geq d_2$ and $|\mu^*| \geq d_1$ due to Lemmata 5 and 6. □

We use this theorem to prove a bound on the approximation ratio of Algorithm 3 for the confluent routing model $>$.

Theorem 12. *Algorithm 3 computes an 8-approximation for the confluent routing model >.*

Proof. According to Theorem 10, for a face $F \in T$ with $\deg(F) = 0$, Algorithm 3 achieves an approximation ratio of 2. Hence a higher approximation ratio can only be reached on parts of outerplanar graphs where $\deg(F) \geq 1, \forall F \in T$. Let us analyze the cost of a deployment produced by Algorithm 3 for this case more precisely. Note that the Breadth-First Tree (BFT) B constructed in Line 1 has n nodes and $n-1$ links, and thus exactly one link per face is missing (each additional link of G creates one of the faces of G). Thus $x = |T| = m - n + 1$ and in the worst case, B has $2x$ leaves, as each missing link can create at most two leaves. Line 4 will be executed exactly x times, for a total deployment cost of $|\mu| \leq 2x + 1 + 2x = 4x + 1$. Since $|\mu^*| \geq \max(2, (x-3)/2)$ according to Theorem 11, it holds for $x > 7$ that $|\mu|/|\mu^*| \leq (8x+2)/(x-3) = \Theta(1)$. □

Finally, to support asymmetric cost models, we suggest to replace the more expensive monitoring node type with the cheaper one greedily, as long as the validity of the deployment is preserved.

4.3 Remark on Symmetric Deployments

Our work also has implications on symmetric deployments. Let $\phi : V \mapsto \{\emptyset, M\}$ be a deployment of symmetric monitoring equipment. We consider G to be monitored iff all edges are on a shortest path according to routing model R between nodes with ME, $\cup\{M^R_{xy} | \phi(x) = \phi(y) = ME\} = E(G)$.

Theorem 13. *Let $\phi^\star$ and $\mu^\star$ be optimal deployments on G for the symmetric and asymmetric case respectively. We have: $|\mu^\star|/2 \leq |\phi^\star| \leq |\mu^\star|$.*

Proof. The upper bound of $|\phi^\star|$ is obtained by defining $\phi' : v \mapsto ME$ iff $\mu^\star \neq \emptyset, \emptyset$ otherwise. Since ϕ' necessarily monitors all the links monitored by $\mu^\star$, it monitors G.

By contradiction assume $|\phi^\star| = |\mu^\star|/2 - 1$. Let $\mu^C : V \mapsto \text{OP} + \text{BC}$ iff $\phi^\star = ME$, $\emptyset$ otherwise. We have $|\mu^C| = 2|\phi^\star| = |\mu^\star| - 2$. Let $e \in E(G)$. Since G is monitored by $\phi^\star$, $\exists a, b \in V(G)$ s.t. $\phi^\star(a) = \phi^\star(b) = ME$ and $e \in SP(a, b)$. Since $\mu^c(a) = \mu^c(b) = \text{OP} + \text{BC}$ and necessarily $e \in M(a, b, \text{OP}, \text{BC})$, e is monitored by μ^c. Since this holds for any $e \in E(G)$, μ^c monitors G with $|\mu^c| < |\mu^\star|$ which contradicts the definition of $\mu^\star$. □

This upper bound is notably tight on trees [4]. Thanks to ϕ's construction, our outerplanar approximation algorithm therefore translates into the first known 16-approximation of symmetric deployments on outerplanar graphs.

5 Related Work

Today we have a fairly good understanding of the requirements to detect and localize single [14, 16] and multiple [7, 8, 9, 17, 24] failures, using symmetric monitoring equipment, also e.g., for building a network tomography infrastructure for the Internet [24]. Some works also go beyond, and aim to estimate the failure severity (e.g., congestion level) [32]. Oftentimes, the algorithms presented in these papers provide probabilistic guarantees, optimizing for the most likely failure event.

To the best of our knowledge, we are the first to rigorously study network tomography for settings where monitor equipment falls into two classes: beacons and observability points. While there have been previous works studying the asymmetric roles of monitoring nodes, e.g., in the context of "multicastbased network tomography" and ''beacon placement problems" [22], we are not aware of any work on the *joint* optimization of the placement of the two tomographic types, under different routing types.

A main focus of our paper is on the routing model. It is known that routing policies substantially restrict which paths are permissible and constrain the effective path diversity [12]. Erlebach et al. [10, 11] study valid s-t- paths and s-t-cuts in the valley-free model and prove the NP-hardness of the node-disjoint min-cut problem. Teixeira et al. [31] study node- and edge-disjoint paths in undirected Internet topology models, but without taking routing policies into account.

There have also been previous works studying different routing models in the context of network tomography, e.g., arbitrary routing [6], routing with cycles [3], and others [23, 25]. In this respect, the paper closest to ours is by Boothe et al. [4] who initiate the study of shortest path monitoring problems similar to ours but in the *symmetric* setting. The authors consider two routing models (union and intersection), and provide optimal algorithms for grid and cactus graphs as well as hardness results for general graphs. Their exact solution for the union model also provides a 2-approximation for our model, in the case that different types of tomography nodes have the same cost. Besides our focus on asymmetry, we extend [4] to additional routing models and develop algorithms for more general graph families. Moreover, we aim to *quantify* the impact of the routing model.

Graph class / Routing model	Tree	Cactus	Outerplanar	General
Union $\cup$	optimal	Alg 2: valid Alg 3: 2-approx	Alg 3: valid	NP-hard Alg 3: valid
Any !	optimal	NP-hard Alg 2+3: valid	NP-hard Alg 3: valid	NP-hard Alg 3: valid
Confluent $>$	optimal	Alg 2: optimal Alg 3: 2-approx	Alg 3: 8-approx	NP-hard Alg 3: valid
Intersection $\cap$	optimal	Alg 2: optimal Alg 3: valid	Alg 3: valid	NP-hard Alg 3: valid

Table 2: Overview of algorithms and complexity results.

6 Conclusion

Table 2 provides an overview of the results presented in this paper. We hope that our formal model and approach can guide the deployment of future tomographic monitoring systems. We consider our work as a first step to understanding the influence of asymmetry in tomographic deployments and of routing models, and there is a wide range of interesting questions for future research. In particular,

it will be interesting to gain deeper insights into the impact of the introduced routing models, also in terms of the offered path diversity. On the technical side, it will be interesting to chart a more comprehensive landscape of the computational complexity and approximability of the underlying optimization problem. In particular, a main open question regards the exact characterization of the graph classes which permit a polynomial-time optimal deployment.

We thank Michael Markovitch for many discussions in the early stage of this work. We are also grateful for the feedback and support from our SIGMETRICS shepherd Paul Barford. This research was partially supported by the Danish Villum project *ReNet* as well as by an Aalborg University talent management grant.

References

[1] Multiprotocol label switching working group, June 2009.

[2] A. Gupta et al. Sdx: A software defined internet exchange. In *Proc. ACM SIGCOMM*, 2014.

[3] S. S. Ahuja, S. Ramasubramanian, and M. Krunz. Srlg failure localization in optical networks. *IEEE/ACM Trans. Netw.*, 19(4), 2011.

[4] P. Boothe, Z. Dvorak, A. Farley, and A. Proskurowski. Graph covering via shortest paths. In *Congressus Numerantium*, 2007.

[5] M. Borokhovich, L. Schiff, and S. Schmid. Provable data plane connectivity with local fast failover: Introducing openflow graph algorithms. In *Proc. ACM SIGCOMM HotSDN*, 2014.

[6] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama. Graph-constrained group testing. *IEEE Trans. Inf. Theor.*, 58(1), Jan. 2012.

[7] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot. Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data. In *Proc. ACM CoNEXT*, 2007.

[8] N. Duffield. Simple network performance tomography. In *Proc. 3rd ACM SIGCOMM Conf. on Internet Measurement (IMC)*, 2003.

[9] N. Duffield. Network tomography of binary network performance characteristics. *Information Theory, IEEE Trans. on*, 52(12), 2006.

[10] T. Erlebach, A. Hall, A. Panconesi, and D. Vukadinović. Cuts and disjoint paths in the valley-free path model of internet bgp routing. In *Proc. International Conference on Combinatorial and Algorithmic Aspects of Networking (CAAN)*, 2005.

[11] T. Erlebach, L. S. Moonen, F. C. R. Spieksma, and D. Vukadinović. Connectivity measures for internet topologies on the level of autonomous systems. *Oper. Res.*, 57(4), 2009.

[12] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Proc. ACM SIGMETRICS*, pages 307–317, 2000.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[14] D. Ghita, C. Karakus, K. Argyraki, and P. Thiran. Shifting network tomography toward a practical goal. In *Proc. ACM CoNEXT*, pages 24:1–24:12, 2011.

[15] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. VL2: A Scalable and Flexible Data Center Network. In *SIGCOMM*, 2009.

[16] J. D. Horton and A. López-Ortiz. On the number of distributed measurement points for network tomography. In *Proc. 3rd ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2003.

[17] Y. Huang, N. Feamster, and R. Teixeira. Practical issues with using network tomography for fault diagnosis. *SIGCOMM Comput. Commun. Rev.*, 38(5):53–58, Sept. 2008.

[18] S. A. Jyothi, M. Dong, and P. B. Godfrey. Towards a flexible data center fabric with source routing. In *Proc. ACM SIGCOMM Symposium on Software Defined Networking Research (SOSR)*, 2015.

[19] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo. *Selected Areas in Communications, IEEE Journal on*, 29(9):1765–1775, 2011.

[20] V. Kotronis, R. Kloti, M. Rost, P. Georgopoulos, B. Ager, S. Schmid, and X. Dimitropoulos. Stitching inter-domain paths over ixps. In *Proc. ACM Symposium on SDN Research (SOSR)*, 2016.

[21] P. Kumar, Y. Yuan, C. Yu, N. Foster, R. Kleinberg, and R. Soule. Kulfi: Robust traffic engineering using semi-oblivious routing. In *arXiv*, 2016.

[22] R. Kumar and J. Kaur. Efficient beacon placement for network tomography. In *Proc. ACM Conf. on Internet Measurement (IMC)*, 2004.

[23] L. Ma, T. He, A. Swami, D. Towsley, and K. K. Leung. On optimal monitor placement for localizing node failures via network tomography. *Perform. Eval.*, 91(C):16–37, Sept. 2015.

[24] L. Ma, T. He, A. Swami, D. Towsley, K. K. Leung, and J. Lowe. Node failure localization via network tomography. In *Proc. ACM SIGCOMM Internet Measurement Conference (IMC)*, 2014.

[25] L. Ma, T. He, A. Swami, D. Towsley, K. K. Leung, and J. Lowe. Node failure localization via network tomography. In *Proc. Conference on Internet Measurement Conference (IMC)*, pages 195–208, 2014.

[26] M. Markovitch and S. Schmid. Shear: A highly available and flexible network architecture: Marrying distributed and logically centralized control planes. In *Proc. IEEE Conf. on Netw. Protocols (ICNP)*, 2015.

[27] S. Moran and Y. Wolfstahl. Optimal covering of cacti by vertex-disjoint paths. *J. Theoretical Computer Science (TCS)*, 84(2):179 – 197, 1991.

[28] Y. Shavitt and E. Shir. Dimes: Let the internet measure itself. *ACM SIGCOMM Computer Communication Review*, 35(5):71–74, 2005.

[29] R. Soulé, S. Basu, R. Kleinberg, E. G. Sirer, and N. Foster. Managing the network with merlin. In *Proc. ACM HotNets*, 2013.

[30] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, 2004.

[31] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker. Characterizing and measuring path diversity of internet topologies. In *ACM SIGMETRICS PER*, 2003.

[32] S. Zarifzadeh, M. Gowdagere, and C. Dovrolis. Range tomography: combining the practicality of boolean tomography with the resolution of analog tomography. In *Proc. 12th ACM SIGCOMM Internet Measurement Conference (IMC)*, pages 385–398, 2012.

7 Deferred Proofs

7.1 Proof of Theorem 7

Due to Lemma 1, the edges of P selected in Line 1 form a connected path or a cycle. In case there is no ME yet, an arbitrary one will be added: no single ME can monitor links. If the routing model allows to monitor the cycle with only two MEs (namely if the cycle length is even and under $\cup$, cf Corollary 1), then the second ME will be inserted diametrically opposed to the first one: C is completely monitored.

From now on, we only consider the case where at least one ME is present, and where 2 ME of each type are required, see Corollary 1. We define $\overline{\mu(v)}$ as OP if $\mu(v) = \text{BC}$, and vice-versa; if $\mu(v) = \text{BC} + \text{OP}$, we can pick any. Let the nodes of the path P of length k be denoted by $l = v_1, v_2 \ldots, v_{k+1} = r$, and let δ refer to the diameter of the cycle. If $P = C$, by convention we set $r = l$ to be the only equipped node. Let $q = |C| - k$ be the size of the cycle part that is already monitored by the current deployment (possibly q is 0). We proceed by proving the theorem separately for three cases depending on q: (i) $q > 1$, (ii) $q = 1$ and (iii) $q = 0$.

(i) $q > 1$: At least two edges of C are monitored already and thus there is a ME on both r and l. There are two sub-cases to consider.

   a) Assume $\mu(r) \cap \mu(l) \neq \emptyset$. Both endpoints of the path contain at least one common ME type. Since $k < |C| - 1$, we have $d(r, v_{\lceil k/2 \rceil}) < \delta$. Therefore for the newly monitored links between r and $v_{\lceil k/2 \rceil}$ by adding the tomography nodes it holds that $\{(v_i, v_{i+1}) | \lceil k/2 \rceil - 1 \leq i \leq k\}$. Since a similar reasoning holds for $d(l, v_{\lceil k/2 \rceil}) < \delta$, both halves of P are monitored with one additional ME. Since no ME deployment can monitor more, C is monitored after Line 4.

   b) Assume $\mu(r) \cap \mu(l) = \emptyset$. A different ME is on each side of the path. If these are the only two ME on C, observe that necessarily two additional MEs will be added, and that the two consecutive executions

of Line 3 will ensure their optimal placement. If there are more MEs on C, let $r' = \arg\min_{c \in \overline{\mu(r)}} d(r', r)$ and $l' = \arg\min_{c \in \overline{\mu(l)}} d(l', l)$. W.l.o.g., assume $r' < l'$. If $r' < m - 1$, we have $d(l, r') > 1$, using the same arguments as in Case (i).a), we conclude that a single ME of type $\overline{\mu(l)}$ is placed at $v_{k-d(r',r))/2}$. Otherwise, if $r' = m - 1$, we have $d(r', l) = 1$, and therefore the rest of the proof follows analogously to Case (ii).

(ii) $q = 1$: If $|C|$ is odd, the deployment is derived analogously to Case (i). We cannot end up in an infinite proof loop, as r' is decremented in each repetition and thus the deployment according to Case (i).a) is assigned in the next argumentation loop for Case (i).b). Otherwise, we have $d(r, v_{\lceil k/2 \rceil}) = d(l, v_{\lfloor k/2 \rfloor}) = \delta - 1$, i.e., two candidate nodes which can monitor the highest number of previously unmonitored links. The actual paths taken by the messages depend on the routing model, and possibly the edge $(\lceil k/2 \rceil, \lfloor k/2 \rfloor)$ is left unmonitored. In this case, the use of an additional ME is mandatory, and it will be deployed optimally during the next loop execution.

(iii) $q = 0$: No edge is monitored yet. If both types are deployed on l, we can use the derivation of case (ii). If the cycle contains only one ME type, three additional MEs are required. Observe that the first execution of Line 3 will add a ME of the opposite type. Hence, after the first execution of Line 4, at least one edge will be monitored. The theorem then follows by Case (i) or (ii). On the other hand, if C contains more than one ME of the same type, we have the following situation. Let $m_a, m_b \in C$ be the nodes with the highest ME distance in-between. If $d(m_a, m_b) > 1$, let v_s be (one of) the midpoint(s) on the shortest path from m_a to m_b, and v'_s the diametrically opposed node. Observe that the first execution of Line 3 will select at least v'_s, which can monitor $|C| - d(m_a, m_b)$ links. The next execution, similar to Case (i).a) will select v_s, monitoring the rest of C. Again, if $d(m_a, m_b) = 1$, there are only two MEs of the same type on C. In such cases, analogously to Case (ii), the deployment depends on C's parity. In the worst case, even if two MEs are present, three additional MEs are required, which is optimal in this case.

In summary, an optimal deployment is found in all subcases, which concludes the proof.

7.2 Proof of Theorem 8

Observe that all BC and OP equipment placed on leaf nodes (i.e., $\in L_0 \backslash C$) is required for the same reason they are required in a tree. Therefore, unnecessary monitoring equipment can only be added due to cycles. When assigning the optimal set of equipment to nexus nodes, *EquipCycle* computes minimal deployments to monitor a cycle thanks to Theorem 7. Thus we have to study the deployment μ' of equipment on nexus nodes. Intuitively, in order to minimize the number of MEs inserted in EquipCycle, one must strive to have the maximum diversity of ME at each nexus node.

If there are no cycles in L_0 (i.e., $C \cap L_0 = \emptyset$ for the set of cycles C), L_0 is optimally equipped. Otherwise, let $C_j \in L_0 \cap C$ be a cycle (a node in the

contracted tree). Since $C_j \in L_0$, C_j has a unique nexus π_u connecting the cycle to the upper layers. Let G_u be the subgraph of G connected to C_j through π_u (we will consider $G_u = \emptyset$ in the special case where $C = \{C_j\}$). If G contains at least another cycle, it is in G_u. For all routing models other than $\cup$, we will necessarily assign at least one OP and one BC in G_u to monitor cycles in G_u. If G contains no other cycle, then G_u is a tree. Observe that due to the depth-first ordering, if G_u has more than one leaf, necessarily $\mu'(\pi_u) = \text{OP} + \text{BC}$ in Line 19. Similarly to the above case, *EquipCycle* will only assign the minimum number of monitoring equipment, two in this case. If G_u has one or zero leaves, their equipment is already optimal, and *EquipCycle* will produce a minimal monitoring, and thanks to Lemma 3 the result will be optimal.

By induction on i, assume that $\cup_{j<i} L_j$ are optimally equipped. Since the only place where monitoring equipment is added are cycles, if $L_i \cap C = \emptyset$, then L_i has no monitoring equipment, which is optimal. Otherwise, let $C_j \in L_i \cap C$. Using similar arguments as for the L_0 layer, observe that all nexus nodes connecting to subgraphs with cycles or two or more leaves are assigned both OP + BC. Let $\pi_1, \pi_2, \ldots, \pi_p$ be the set of C_j's nexus nodes in clockwise order ($>_d$) starting from an arbitrary nexus node such that $k > l \Leftrightarrow \pi_k >_d \pi_k$. Let $d_1 = d^+(\pi_1, \pi_2), d_2 = d^+(\pi_2, \pi_3), \ldots, d_p = d^+(\pi_p, \pi_1)\}$, where here d^+ is the hop distance in C_j using only the direction imposed by the clockwise order $>_d$. Let ℓ be half the cycle length: $\ell = \lfloor 1/2 \sum_{k=1}^{p} d_k \rfloor$, and let $m = \arg\max_{1<k<p} d_k$.

1. *Case $d_m > \ell$:* We know that the shortest path between any monitoring equipment in G_m and $G_{m+1[p]}$ will not visit the cycle nodes between π_m and $\pi_{m+1[p]}$ in the depth-first search (DFS): additional monitoring equipment in the cycle is required. If $p > 2$, we know by construction that we have at least two members of one monitoring equipment type (say, 2 BCs: s_1 and s_2). Thanks to the DFS leaf ordering, we also know that there exists at least one OP: o_1 such that $d^+(s_1, o_1) < \ell$ and $d^+(o_1, s_2) < \ell$. Since $d_m < \sum_{k=1}^{p} d_k$, putting the only required OP on a node $v \in C_j$ such that $d^+(\pi_m, v) = \lceil 1/(2d^+(\pi_m, \pi_{m+1[p]})) \rceil$ optimally solves the problem. If $p = 2$ and $|C| = 1$ C_j is on the only cycle of G that has two leaves. Thanks to the DFS ordering, there is both an OP and a BC assigned virtually to the nexus nodes. Hence it is necessary to add both an OP and a BC, for instance at location v: $d^+(\pi_m, v) = \lceil 1/(2d^+(\pi_m, \pi_{m+1[p]})) \rceil$: G is optimally monitored with four MEs as it contains a cycle. If $p = 2$ and $|C| > 1$, at least one nexus node connects to the two monitoring equipment types. Assume the second nexus node is connected to a BC, then adding an OP at the same node v solves the problem optimally.

2. *Case $d_m < \ell$:* Shortest paths from one nexus node to another naturally explore all edges of C_j. To know which equipment to add (if any), multiple cases need to be considered. If $p > 4$ or $p = 3 \wedge |C| > 1$, all the required monitoring equipment is already in the network, and since BC and OP alternate around the cycle: C_j requires no additional monitoring equipment, which is optimal. Since we cannot have $p = 2 \wedge d_m < \ell$, the only case to handle is $p = 3 \wedge |C| = 1$. In this case, we are processing G's only cycle that has 3 nexus nodes. If the graph has at least 4 leaves, π_1 contains both an OP and a BC, and whatever π_2 and π_3 contain, will allow us to monitor C_j optimally (without any addition). If G has exactly 3

leaves (it cannot have fewer leaves since $p = 3$), we need to add exactly one element to monitor the cycle. Assume π_1 has a BC, then necessarily π_3 too. Adding an OP between those nexus nodes monitors the cycle, and thus monitors C_j optimally (four elements for a graph with a cycle). This concludes the induction step, and consequently G is monitored optimally by μ computed by Algorithm 2.

3. *Otherwise:* If $d_m = \ell$, one of the two cases discussed above applies, depending on the routing model: if the route connects two nodes w.r.t. clockwise order: $d_m < \ell$, otherwise $d_m > \ell$.