Implications of Routing Coherence and Consistency on Network Optimization

Yvonne-Anne Pignolet[1] Stefan Schmid[2] Gilles Tredan[3]
[1] DFINITY, Switzerland [2] Faculty of Computer Science, University of Vienna, Austria [3] CNRS-LAAS, France

***Abstract*—In network optimization problems, from traffic engineering to network monitoring, the routing model is typically considered as something given and fixed. This paper is motivated by the fundamental question how the ability to *change* and *optimize* the routing model itself influences the efficiency at which communication networks can be operated. To this end, we identify two main dimensions of the routing model: *consistency* (of a single route) and *coherence* (of sets of routes). We present analytical results on the impact of the routing model on the achievable route diversity as well as on the runtime of solving optimization problems underlying different case studies. We also uncover that it can sometimes be beneficial to *artificially* restrict the routing model, to significantly reduce the computational complexity without negatively affecting the route diversity much.**

I. Introduction

While most communication networks feature a routing mechanism which supports the delivery of packets from their source s to their destination t across a multi-hop network, they can differ significantly in how and to what extent the routes taken by packets can be controlled. Routing models have evolved dramatically over the last years, and indeed, more flexible routing models have been a main driver behind recent innovations in networking [11]. At one end of the spectrum lie traditional networks in which packets are routed along shortest paths and in a *(IP prefix) destination-based* manner (using protocols such as, e.g., OSPF). Such networks often provide only limited flexibility: destination-based routes are confluent, and flows towards the same destination remain on the same route once they meet. Furthermore, the only knob available to the operator to influence the routes is to set the *link weights* based on which the shortest paths are selected. In the late 1990s, increasing traffic volumes and the need for a more reliable performance led to the design of more advanced routing technologies (traffic engineering). MPLS enabled a *per-flow* traffic engineering, supporting the definition of more general (not necessarily shortest) routes, and SDNs and OpenFlow facilitated a *direct* control over the forwarding tables. In particular, besides forwarding, SDN and OpenFlow also support a more general *modification* of packet headers, enabling more advanced routing services: packets may are not only forwarded along simple paths, but routes may have *loops*.

But not only the technology and specific protocols can influence the flexibility at which routes can be selected, but also the network policy. For example, inter-domain routes in the Internet are often driven by business/financial aspects and may be *valley-free*. There also exist policies which force traffic to be routed via certain waypoints (e.g., a middlebox for deep packet inspection, a firewall, or a WAN optimizer) between source and destination [17], or to *explicitly avoid* such waypoints (e.g., routing via certain countries or via certain network elements such as route reflectors). Thus the routes taken by packets can depend on aspects beyond the pure network topology.

This paper is motivated by the observation that while it is intuitively clear that different routing models have an impact, we lack a systematic comparison of existing and future routing models, not only with respect to the routing flexibility they provide (the "route diversity"), but also with respect to the cost resp. quality and time complexity at which network optimization problems can be solved in these different routing models. Indeed, as we show in this paper, and complementing the perspective usually taken in the literature, many aspects of network optimization to a large extent does not only depend on the complexity of underlying network topology but also on the routing model used.

Our main ***contributions*** are a systematic study of the impact of the routing model on fundamental network optimization problems. To this end, we first propose a taxonomy of routing models, along two dimensions, *consistency* and *coherence*:

1) *Consistency* (Π)*:* Consistency relates to properties of a single route, e.g., some routing models require shortest or loop-free paths, or policies may dictate that packets should (not) go through a certain waypoint.
2) *Coherence* (Σ)*:* Coherence refers to constraints on how multiple routes relate to each other. For example, packets must travel confluent routes towards the same destination.

We illustrate how the consistency and coherence influence the diversity of routes on a given network and propose a hierarchy of coherence models which impacts the achievable performance. Furthermore, we demonstrate how routing algebra properties can impact coherence. We consider two canonical problems as case studies. (i) *Traffic Engineering:* A main goal of Traffic Engineering is to keep load low, thus we consider the impact of routes on network load in this case study. (ii) *Monitoring:* Dependable communication networks require (automated) monitoring, e.g., to check the availability of links or entire routes. The objective in this case is hence to keep track of the network state at low cost. Moreover, we show how routing models can help to prove new properties using OSPF as an example.

ISBN 978-3-903176-28-7© 2020 IFIP

Our approach leads to several interesting observations. For example, we find that the same optimization problem on the same network can be polynomial-time solvable under one routing model and NP-complete under another routing model. This uncovers an optimization opportunity: by restricting the routing model *artificially*, i.e., by introducing constraints on the routing which do not negatively affect the route diversity by much, the computational complexity may be reduced significantly.

II. A Taxonomy of Routing Models

This section first introduce a basic network model and then presents and discusses a taxonomy to classify routing protocols.

A. Preliminaries

We consider a basic model in which the network is represented as an undirected connected graph $G(V, E)$, with $n = |V|$ devices (nodes) and $m = |E|$ communication links connecting them. Nodes and links may have attributes representing costs, constraints, etc assigned to them. An (s, t)-*route* is a sequence of k nodes $r = (v_1 = s, v_2, \dots, v_k = t)$ such that $(v_i, v_{i+1}) \in E$. Note that a route r does not necessarily have to follow a shortest path and not even a simple path: the route may contain *loops* (e.g., in order to visit a certain waypoint, such as a firewall, along the route), and hence, in graph theory terminology, the route forms a *walk*. However, for ease of presentation, we first focus on simple (but not shortest) paths, and write $r[v_i, v_j]$ to denote the subsequence of r between nodes v_i and v_j. Section II-E describes how to extend our model to routes with loops.

Given two routes r_1, r_2, we denote by $r_1 \cap r_2$ the set of links appearing in both routes. Furthermore, we consider the graph induced by the union of two or more routes, ignoring the order of the links: thus $r \cup r'$ implies an induced network $G(V, E)$, where $V = \{v_1, \dots, v_k, v'_1, v'_{k'}\}$ and $E = \{(v_i, v_{i+1}) | (v_i, v_{i+1}) \in r \ \vee \ (v_i, v_{i+1}) \in r'\}$.

We refer to the set of all possible routes which differ by at least one edge between any two nodes $x, y \in V$ in G by $R(x, y)$. In addition, we denote by $R(x, .)$ (resp. $R(., y)$) the set of all routes that start at node x (resp. finish at node y). Finally, to emphasize when we consider *all* routes in a network G (not only between certain endpoints), we write $R(G)$.

Given a set of source-destination pairs, a *routing algorithm* produces a set of routes for these pairs. In this paper we do not focus on the algorithms constructing these routes but rather on the properties the resulting route sets exhibit: Hence the routing algorithms are abstract providers of routes. This allows us to first compare all algorithms on a generic basis, and second to classify algorithms using properties on their route sets that are useful from a theoretical and practical perspective.

Definition 1 (Route Set $\mathcal{S}$)**.** *A* route set $S \subset R(G)$ *contains zero, one or several routes* $r_i \in R(G)$ *for each source-destination pair.*

Consistent with this approach, when we study properties of routing models, we will distinguish between *blackbox* properties which can be checked by only observing the routes themselves, as opposed to *whitebox* properties that require additional knowledge on the "infrastructure", e.g., on the network topology, link weights, algorithm parameters, etc.

B. Dimension 1: Consistency

The first dimension of our taxonomy is consistency: a property defined on *individual route* of a route set. Formally, a route r on a graph G is Π-consistent if the predicate $\Pi(r, G)$ defined over the links and nodes of the route r on G evaluates to true. For example, a basic consistency predicate is that all routes of a route set are of minimum length: shortest path routing is employed by well-known protocols, e.g., OSPF, which ensure that all flows are routed along shortest paths with respect to edge lengths (routing weights). Other examples of consistency properties are related to the policy-compliance of a given flow, e.g., ensuring that a route did traverse certain waypoints, did *not* traverse blacklisted parts of the network, or conforms to business relationships (like valley-freedom).

In general, a consistency predicate Π can be used as a filter: only a subset of all possible routes $R(x, y)$ between x and y may fulfill Π. In the following, let $R_\Pi(x, y) \subset R(x, y)$ denote the set of routes from x to y that are consistent w.r.t. Π. We write $R_\Pi(G)$ and call it the set of all Π-*consistent routes* of a graph G.

Note that many of the consistency predicates used in practice depend on properties of the elements of the underlying network *infrastructure*, such as waypoints, edge type classes or weights representing the cost or latency incurred when using them. In particular, predicates such as "is a shortest path" even require additional information about the infrastructure, beyond the links of the current path: in order to be able to verify that this path is indeed the shortest between a given source and a given destination, we need to know the alternative links (and their weights) in G. However, there are also consistency properties which do not require such additional information, for example a predicate of the form "the route length is at most ℓ" or loop-freedom. We will refer to consistency properties which depend on the infrastructure as *whitebox* consistency properties, and to consistency properties which do not require such information, as *blackbox* consistency properties.

Consistency properties (e.g., valley-freedom, waypoint routing, multipathing, etc.) are often described using regular languages [24], [17], [16], e.g., over labels on links and nodes: it is required that all valid routes in the graph adhere to this regular expression. For example, $s.^*w.^*t$ could express that a route from s to t should traverse a waypoint w. Or $(c2p)^*(p2p)?(p2c)^*$ could express a valley-free routing policy where edge labels are used to denote peer-to-peer ($p2p$), provider-to-customer ($p2c$), or customer-to-provider ($c2p$) relationships [17].

Another approach, based on algebraic methods, considers routing policies as a function that selects, from the set of all paths from a source to a destination, preferred paths according to predefined rules. Simply put, a routing algebra defines a set of "legal" or policy-compliant routes. This definition is broad enough to contain many routing policies, e.g., shortest paths, widest path, most reliable path, widest-shortest path, shortest-widest path, valley-free paths, etc. A large body of literature analysed routing protocols in such a framework, e.g., [23], [22], [15], [2] to name but a few.

The crucial components of a *routing algebra* are a partially-ordered commutative semi-group with a compatible infinity element: $A = (W, \phi, \bigoplus, \preceq)$, where W is the set of possible edge weights (i.e., different edges can have different costs), ϕ ($\phi \notin W$) denotes an infinity element assigned to unusable edges/routes, and $\bigoplus$ is a composition operator for weights (e.g., latency related edge costs add up while bandwidth-related edge costs are naturally subject to min/max operations). Given a route we obtain its weight by combining the weights of its constituent edges with $\bigoplus$. $\preceq$ is a partial order for weight comparison of edges and routes. A *preferred route* in the algebra A between two nodes is one with the smallest weight according to $\preceq$. The infinity weight ϕ is compatible with $(W, \bigoplus)$ according to $\preceq$ in the sense that it is absorptive $w \bigoplus \phi = \phi$, $\forall w \in W$ (a route with contains an unusable edge is unusable), and maximal $w \prec \phi$, $\forall w \in W$ (any route without an unusable edge is preferred over a route with an unusable edge).

To give an example, shortest path routing, where valid paths between two nodes minimise the sum of the weights of its constituent edges, corresponds to the algebra $(\mathbb{R}+, \infty, +, \leq)$, where positive edge weights ($\mathbb{R}+$) may describe a property like the latency or cost of this edge, which is added up ($+$) along a route, and shorter routes are better ($\leq$). Widest-path routing prefers paths which have the largest bottleneck capacity, i.e., $(\mathbb{R}+, 0, \min, \geq)$ where positive edge weights ($\mathbb{R}+$) describe the link bandwidth, the total bandwidth provided along a route is the minimum ($\min$) offered on any of its constituent links, and wider paths are better ($\geq$).

C. Dimension 2: Coherence

The second dimension concerns the coherence $\Sigma(R, G)$ of route sets R produced by a routing algorithm. Similarly to the above, we denote by $\mathcal{R}^\Sigma$ the set of route sets that fulfill a coherence predicate Σ. Since Σ describes a relationship between multiple routes, $\mathcal{R}^\Sigma$ is a *set of sets*: each set of routes $R \in \mathcal{R}^\Sigma$ satisfies the coherence predicate Σ.

Note that a coherence predicate compares multiple routes to each other (e.g., if and in which nodes and links they are the same or different). Many important coherence properties do not require references to the infrastructure network G (i.e., they are *blackbox* coherence properties). Take for example destination-based routing, which can be expressed generally as "once two routes towards the same destination meet, they will follow the same route from then onward". Indeed, as we will see, when describing coherence properties, it often matters when two routes *meet*.

There are also *whitebox* coherence properties, which require knowledge about the infrastructure. For example, consider a network which includes two waypoints w_1, w_2 of the same type (e.g., an intrusion detection system). A coherence property may require that two flows, one from s_1 to t_1 and one from s_2 to t_2, either *both* go through w_1 or *both* go through w_2.

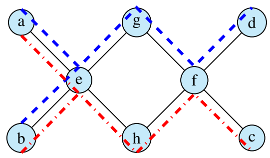

Fig. 1. Example: The routes from a and b to c and d are confluent: for each destination they follow the same route once they meet. The routes are not contained as paths between e and f differ.

Some natural coherence properties are the following:

Definition 2 (Basic Coherence Property Examples)**.** *Let G be a graph and Σ a coherence predicate. Basic coherence properties include:*

(i) ***Multi*** *($\star$): In this model, an arbitrary subset of $R(G)$ is valid: for each set of routes $R \in \mathcal{R}^*$, it holds that $R \subseteq R(G)$. In particular, more than one route between a source and destination node may be included in R.*

(ii) ***Any*** *(!): In this model, we only require that at most one route between any source-destination pair exists, and there are no other constraints on the route: for each set of routes $R \in \mathcal{R}^!$, it holds that $|R(x, y)| \leq 1$ $\forall x, y \in V(G)$.*

(iii) ***Confluent*** ***($>$):*** *In the confluent model, the route choice at each node is source-invariant. I.e., the next hop is determined by the destination. Let $R \in \mathcal{R}^>(G)$ $\forall w, y \in V, \forall r, r' \in (R \cap \mathcal{R}(., y))$ it holds that $w \in r \cap r' \Rightarrow r[w, y] = r'[w, y]$. Note that such an element $w \in r' \cap r$ has to carry the same inport and attribute in both sequences. This also holds for all elements in $r[w, y]$ and $r'[w, y]$.*

(iv) ***Contained*** ***($\subseteq$):*** *In the contained model, any two routes share at most one contiguous subsequence $\forall z, w \in V, \forall r, r' \in R$ it holds that $\{z, w\} \in r \cap r' \Rightarrow r[z, w] = r'[z, w]$.*

(v) ***Forest*** *(T)* ***and Graph*** *($\mathcal{G}$):* *In the most constrained model, the union of* all *routes in any set $R \in \mathcal{R}^T$ is a forest. More generally, the coherence restriction could be extended $R \in \mathcal{R}^\mathcal{G}$ for other graph classes, e.g., DAGs or planar graphs.*

(vi) ***Symmetric Routing*** *($\leftrightarrow$): A set of routes R is symmetric if it holds for all source-destination pairs (s, t), if a route r from s to t is in R then the reverse route from t to s is in R too.*

Note that these coherence properties differ in terms of the path subsequences shared by the different routes. E.g., $\star$ and ! do not have any constraints on shared subsequences, while $>, \subseteq, T, \leftrightarrow$ require shared subsequences to adhere to rules. In *symmetric* routing, the same routes between two nodes are used in both directions. It is easy to see that for example routes adhering to T routing models are always symmetric. For an example where the confluent $>$ and the contained $\subseteq$ routing models are different, see Figure 1 with valid confluent routes which are not contained.

D. Combining Consistency and Coherence

There can hence be two different types of restrictions on a route set: related to consistency Π and related to coherence

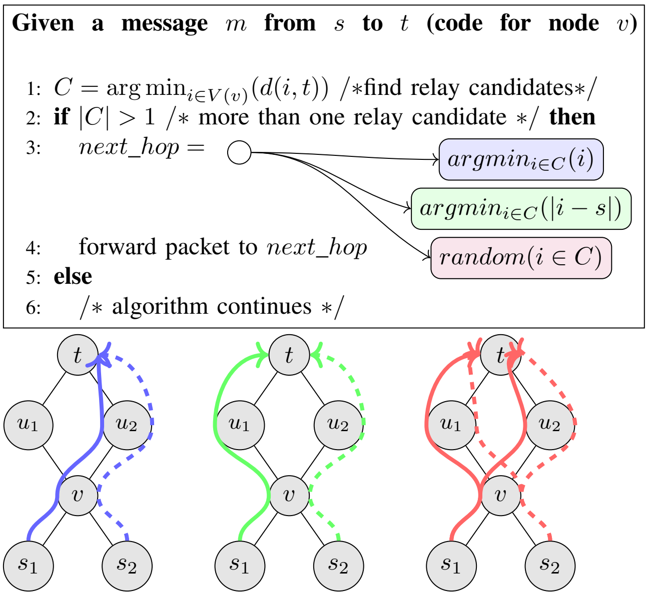

Fig. 2. Prototype of a shortest path routing algorithm, and impact of the tie-breaking. Produced routes are always consistent. Depending on the nature of the tie breaking, the set of produced routes will have different properties. In the blue example, routes will always be confluent, whereas in the green examples, the produced route set might be *Any* (!). A non-deterministic tie-breaking like in red may produce *Multi* ($\star$).

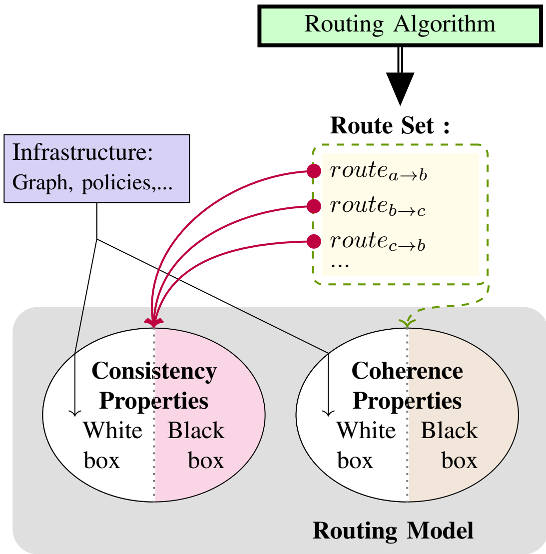

Fig. 3. Overview of our taxonomy: A routing algorithm produces a set of routes. Standard approaches mostly focus on the properties of routes taken *individually*, which we refer to as consistency properties. Conversely, coherence relates to properties of the set of routes taken as a whole. For each of these properties, we distinguish black box properties (that can be checked without infrastructure knowledge, e.g. loop-freedom) and white box properties (that refer to a specific topology, e.g., shortest paths). The combination of consistency and coherence properties defines the routing model.

Σ. Both limit the classes of route sets $R_{\Pi}^{\Sigma}(G)$ through a given network G. For any set of routes $R \in R_{\Pi}^{\Sigma}(G)$, the routes R *jointly* fulfill Σ, and each route *individually* fulfills Π. For many networking problems $R_{\Pi}^{\Sigma}(G)$ serves as a better class definition when reasoning about network algorithms (e.g., for traffic engineering, monitoring, etc.) than the network topology or consistency and coherence properties individually (as discussed in more details later). Motivated by our observations, we will define a route set to adhere to a routing model as follows.

Definition 3 (Routing Model $\mathcal{M}$). *Let* Π *be a consistency criterion and* Σ *a coherence criterion. The* routing model $\mathcal{M}_{\Pi}^{\Sigma}(G)$ *consists of all route sets* S *that satisfy* Π *and* Σ*, i.e.,* $S \subset R_{\Pi}$ *and* $S \in \mathcal{R}^{\Sigma}$.

The fewer constraints we have on coherence and consistency for a routing model, the higher the number of schemes satisfying the predicates. In other words, less constrained models $\mathcal{M}_{\Pi}^{\Sigma}$ contain route sets of larger size and more route sets.

In Figure 2 we illustrate the connection between a routing algorithm and the resulting routing model. It presents a partial prototype of a shortest path routing algorithm: line 1 ensures that paths are selected according to this consistency criteria. However, depending on the contents of line 3, the resulting routing model might end up having various coherence properties. Below, we depict the consequences of 3 example implementations, together with the possible resulting routes.

This example demonstrates the importance and impact of tie-breaking in such protocols. Other protocols may not fit this prototype algorithm, yet produce route sets that obey the same logic. Let us also underline that many routes are affected by line 3, for instance in a regular hypercube there are 2^{k-1} shortest paths to nodes at distance k: the actual route will be selected through $k-1$ successive evaluations of line 3.

E. Generalization for Routes with Loops

Our definitions can easily be extended beyond simple paths. In this case, we do not only have to account for the current node and destination, but also have to consider the *packet's state*, e.g., a flag denoting whether a packet is on its route "before or after the waypoint", as well as the *router's state* (e.g., counters). These states can be modelled as additional *attributes*, which can be matched in forwarding rules of routing protocols. Analogously, the resulting routes can be annotated with these attributes. Thus the notion of two routes "meeting" at a node v in our taxonomy needs to be generalised to refer to these annotations. I.e., two routes meet if in addition to visiting the same node they feature the same attributes at this node. Another way of looking at this, is to consider a node v occurring on routes with attributes a_1 and a_2 as two different instances of a node, v_1 and v_2 respectively. On the multi graph induced by the set of annotated nodes and links the routes are thus loop-free and the original definitions can be used.

F. Summary and Taxonomy

In summary, we propose to study sets of routes as the generic consequence of any routing algorithm. This allows us to focus only on the consequences of these algorithms in our taxonomy (Figure 3), regardless of the internal logic that led a given algorithm to produce a particular set of routes.

Inside this model, we identify two canonical categories of properties to describe those route sets: *(i)* consistency properties, describing properties satisfied by each individual route of the considered set (e.g., properties that paths are *simple*, *shortest*, or at most k hops long). And *(ii)* coherence properties, describing properties satisfied by the route set as a hole (e.g. the routes are *symmetric*, *confluent*). We can further distinguish two types of coherence properties. First, internal coherence properties, that can be expressed using only elements of the route set (e.g., the route set is *contained*) and can be seen as the counterpart of consistency (expressed as predicates involving a single route against elements of the infrastructure). Second, generalized coherence properties, that can only be expressed using predicates involving both multiple routes and elements of the infrastructure. We can then define the routing model as the combination of consistency and coherence properties fulfilled by a route set. We illustrate in Figure 2 how a simple shortest path routing algorithm (designed with a predefined consistency criteria) can provide route sets belonging to different routing models because of the different coherence properties induced by its tie-breaking behavior.

III. General Analysis and Implications

This section presents an analysis of the impact of the routing model, based on our taxonomy, namely route diversity, hierarchies and the interdepence of routing algebra properties and coherence.

A. Notions of Route Diversity

Given our taxonomy, we can refine the intuitive notion of "path diversity". First, the term *route diversity* is more accurate to represent the flexibility offered by a variety of routes between a source and destination since routes do not necessarily have to follow simple paths and may contain loops, as discussed earlier. Route diversity can come in different flavors.

If we consider consistency only, we can define $(s,t)_\Pi$-*route-diversity* to count the number of different Π-consistent routes a packet travelling from source s can take to reach its destination t on a graph G. E.g., there might be several shortest paths between s and t, or several valley-free paths. The higher this number, the more distinct Π-consistent route sets exist.

For a given route set R, we can define the (s,t)-*route-set-diversity* to be $|R(s,t)|$, the number of distinct routes between s and t in R. We further define (s,t)-*subsequence-route-set-diversity* as the number of different routes a packet travelling through s and t can take, according to a route set R (regardless of the source and destination of packets), i.e., $|\{r|r \in R \ \wedge \ len(r[s,t]|) > 0\}|$.

Note that the subsequence-route-set-diversity definition is more general in the sense that depending on a routing model certain paths between s and t may only be traversed by packets emitted by s' and not by packets originating at s. To indicate the complexity and quality of some problems one of the two may be more appropriate. E.g., for the monitoring problem described in our case study (see Section IV) the subsequence-route-diversity of a routing model matters. This is due to the fact that a route measurements can be used to infer metrics of the links they contain and thus a $r[u,v]$ subsequence of a (s,t)-route r traversing u and v can monitor links in $r[u,v]$, even though there might be no (u,v)-route that contains $r[u,v]$.

To analyze the impact of Π-consistency and Σ-coherence not just on a pair of nodes and of single route set but on the number of route sets that conform with Π and Σ , we can define the *routing model diversity* as $diversity^{\Sigma}_{\Pi} := |\mathcal{M}^{\Sigma}_{\Pi}|$, the number of route sets that adhere to $\mathcal{M}^{\Sigma}_{\Pi}$.

To illustrate how the routing model affects route diversity, let us consider a fundamental example, depicted in Figure 4. Let S be the set of route sets with routes between a set of s sources $s_1, \dots s_s$ and two destinations t_1 and t_2. In this example, the blue nodes connect sources and destinations by k parallel routes of the same length. The number of possible loop-free !-coherent route sets is thus $diversity^{!}_{\oslash} = k^{s+2}$, $diversity^{>}_{\oslash} = diversity^{\subseteq}_{\oslash} = k^2$ for the confluent and contained routing model and $diversity^{T}_{\oslash} = k$ for forest-coherent routing. The first observation drawn from these results concerns the restricting power of routing models. While the network exhibits a number of loop-free route sets growing exponentially with the number of sources in the ! model, this combinatorial explosion is no longer possible under more restrictive routing models. The impact of the topology on those restrictions can be seen very well on outerplanar graphs: the number of tree route sets is still exponential in the number of faces (which in turn can be as large as $\Omega(n)$).

Similarly to the coherence example above, consistency influences the route model diversity. Consider two nodes s,t on an odd cycle. Under shortest path consistency there is exactly one route between s and t possible and thus every !-coherent route set is also $>$-coherent. A more relaxed consistency criteria may allow two routes. In this case, we could construct an !-coherent route set that is not confluent by adding the long routes for the two neighbors of t.

B. Coherence Hierarchy

We first observe that some of the properties defined in the previous section form a hierarchy of increasingly flexible routing. In particular the number of possible route sets $\mathcal{R}^{\Sigma}_{\Pi}(x,y)$ between two nodes x and y (or on G in general), depends, besides the topology, on the routing model defined by Σ and Π.

In the following, we prove the *hierarchy of coherence* (for any consistency property Π as it affects all sets the same way).

Theorem 1. *Let G be a graph. We have $\mathcal{R}^{T}_{\Pi}(G) \subseteq \mathcal{R}^{\subseteq}_{\Pi}(G) \subseteq \mathcal{R}^{>}_{\Pi}(G) \subseteq \mathcal{R}^{!}_{\Pi}(G) \subseteq \mathcal{R}^{\star}_{\Pi}(G)$, for any consistency Π.*

Proof. To improve readability, we omit the subscript Π in the proof. Only routes that satisfy Π are considered in the following. *Containment*: Let $S \in R^T(G)$ be the union of all routes, forming a tree. In particular all routes that pass through a particular node w form a tree and thus at most one contiguous subsequence for each pair of routes, hence $S \in R^{\subseteq}(G)$.

Let $S \in R^{\subseteq}(G)$ and let w be a node. All pairs of routes containing w have at most one contiguous subsequence. This holds in particular for routes destined for w: $S \in R^{<}(G)$.

Let $S \in R^{<}(G)$. By contradiction assume that there exists x, y such that $|R(x,y) \cap S| > 1$. Let r and r' two such routes. Since r and r' are in $R(.,y)$, we know that $r \cup r'$ do not split after they meet. Since $x \in r \cap r'$, we conclude that $r[x,y] = r'[x,y] = r = r'$.

We note that if the network is a tree, all models become the same (*possible equality*). We construct an example where all those coherence sets are equal. Let T be a tree, and Σ a classic "shortest path" consistency criteria. Observe that $\forall S \in R^{\star}(T), \bigcup_{r \in S} r$: we have $R^{\star}(T) \subseteq R^{T}(T)$, from which we conclude $R^{\star}(T) = R^{T}(T)$, settling the case for the intermediary models $!, <$ and $\subseteq$. $\square$

The tree example in the proof shows that topologies have an impact on the route diversity. More precisely, a greater link density allows for many routes, and hence allows for many route combinations that populate the routing models hierarchy.

Intuitively, a higher route diversity allows for more possible configurations, some of which may provide more desirable output. This intuition follows from Theorem 1.

Corollary 1. *For any optimization problem, let* $\text{qual}(S, G)$ *denote the quality measure of the best solution achievable on a graph* G *for a given route set* S*. By extension, let* $\text{qual}(R^{\Sigma}_{\Pi}, G) = \max_{S \in R^{\Sigma}_{\Pi}}(\text{qual}(S, G))$*. Then* $\text{qual}(R^{T}_{\Pi}(G), G) \leq \text{qual}(R^{\subseteq}_{\Pi}(G), G) \leq \text{qual}(R^{>}_{\Pi}(G), G) \leq \text{qual}(R^{!}_{\Pi}(G), G) \leq \text{qual}(R^{\star}_{\Pi}(G), G)$.

Proof. Consider $qual(R^{T}_{\Pi}(G), G)$ and $qual(R^{\subseteq}_{\Pi}(G), G)$, and let $S_1 \in R^{T}_{\Pi}(G)$ and $S_2 \in R^{\subseteq}_{\Pi}(G)$ be (one of) the route sets realizing this optimum. Since $S_1 \in R^{T}_{\Pi}(G) \subseteq R^{\subseteq}_{\Pi}(G)$, we have in particular $S_1 \in R^{\subseteq}_{\Pi}(G)$ and therefore $qual(S_1) \leq \max_{S \in R^{\subseteq}_{\Pi}(G)}(qual(S, G)) = qual(R^{\subseteq}_{\Pi}(G), G)$. This settles the case for $R^{T}_{\Pi}(G)$ and $R^{\subseteq}_{\Pi}(G)$. The inequalities for the other cases can be derived analogously. $\square$

C. Equivalence Under Symmetry

If routes are symmetric (using the same links in both directions), confluent routes are contained.

Theorem 2. *A symmetric and confluent routing model ensures that valid routes are contained.*

Proof. For the sake of contradiction, assume the opposite for simple paths, i.e., two valid routes between nodes e and f, where $a - h \in V$ s.t. $g \in R(e, f)$; nodes e, g, f are incident to links in $R(a, d)$, node g is not on any route in $R(b, c)$, and e, f are on routes in $R(b, c)$, e.g., like in Fig. 1. Consider $T_f = \cup_{v \in V} R(v, f)$, the confluent tree leading to f. Since T_f is a tree, and since $g \in R(e, f) \wedge (g, e, f) \in R(a, d)$ we deduce $g \in R(f, a)$ (or $g \in R(f, e)$ w.l.o.g.) due to symmetry. Since $T_b \cup_{v \in V} R(v, b)$ is also a tree, we have $R(f, b) \subset R(e, b) \subset R(c, b) \subset T_b$. Thus $g \in R(c, b)$ and due to symmetry we have a contradiction. The same argument can be generalized for walks using inports and attributes instead of loop-free paths. $\square$

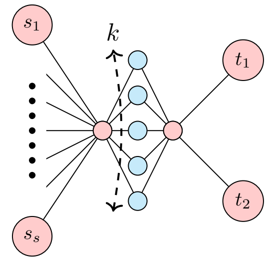

Fig. 4. Graph illustrating the impact of the choice of routing policies on the route diversity. The route diversity in this graph is high for some models and low for others, implying a high variance in network problem solution quality for different routing models.

D. Routing Algebras Can Impact Coherence

As mentioned earlier, a routing algebra can be used to describe consistency properties: for each source-destination pair we can determine whether a route is preferred. Yet most routing algebras do not have an impact on coherence properties of sets of routes. However, some classes of routing algebras can be used to make statements about the kind of coherence properties a route set may *not* be able to satisfy. For example, results from [23], [22] can be interpreted with respect to coherence properties. To this end, we define the set of routes that represent all preferred paths of an algebra A by $\mathcal{R}_A$. An !-coherent route set $S \subseteq \mathcal{R}_A$ *derived* from A contains at most one route per source-destination pair. Furthermore, we need the definition of *regular* routing algebras, which feature a total order $\preceq$ and satisfy *monotonicity*: $w_1 \preceq w_2 \bigoplus w_1, \ \forall w_1, w_2 \in W$ and *isotonicity*: $w_1 \preceq w_2 \implies w_3 \bigoplus w_1 \preceq w_3 \bigoplus w_2, \ \forall w_1, w_2, w_3 \in W$. Monotonicity requires that prepending an edge (or path) of weight w_1 to another edge (or path) of w_2 can only make it less preferred. By commutativity, the same applies to appending edges/paths. Isotonicity, on the other hand, requires $\preceq$ to be compatible with the semigroup $(W, \bigoplus)$ in the following sense: if an edge/path is preferred over some other one, then prepending or suffixing both with a common edge or path maintains this relation. As an example, BGP and IGRP can both be represented by routing algebras. BGP is regular while IGRP is not isotonic and thus not regular [23]; a fact that illustrates how the above definitions can classify real-world routing policies. It also follows follows from [23] that an algebra A can be implemented by a destination-based routing function on any graph, if and only if A is regular. These results imply that a $\mathcal{R}_A$ representing an algebra A can be turned into a confluent route set by removing some of the routes. In other words, a routing algorithm which selects one next hop based on the destination only and which conforms with algebra A, can exist if and only if A is regular and always produces a confluent route set. In other words, we can make the following observation.

Observation 1. *A routing algorithm that turns a preferred route set* $\mathcal{R}_A$ *into a* !*-coherent route set* $\mathcal{R}' \subseteq \mathcal{R}_A$ *which is* $confluent$ *on every network exists if and only if* A *is regular.*

Other aspects that have been studied for routing algebras are their scalability and memory requirements [23], [22]. Route sets that do not represent a regular algebra are incompressible in the sense that their policy does not scale well, as the memory needed to store the local routing process of some node increases with the number of nodes in at least one network topology.

Observation 2. *Given a non-regular routing algebra A, there are networks where no !-coherent route set derived from $\mathcal{R}_A$ for all source-destination pairs can be confluent.*

On the other end of the spectrum, Retvari et al. [22] prove that if and only if a route set represents a monotonic and *selective* algebra, i.e., $w_1 \bigoplus w_2 \in \{w_1, w_2\}$ for each $w_1, w_2 \in W$, this route set adheres to tree coherence and is thus highly compressible.

Observation 3. *A route set $\mathcal{R}_A$ representing a monotonic and selective algebra A is T-coherent.*

E. Summary

To summarize our observations, coherence and consistency can have a large impact on the number of possible route sets on a given graph. As many networking problems involve exploring the space of possible route sets to find an optimal one, the impact of the routing model on the structure of this "potential solutions space" is twofold. First, by restricting the size of the solution space, constraining routing models can forbid the most optimal solutions (see Corollary 1). Second, by changing the nature of the solution space (e.g., its size), coherence also impacts the time complexity of algorithms, as well as the cost or performance of the solution. The next section provides examples that instantiate these differences on specific algorithmic problems.

IV. Implications for Fundamental Case Studies

So far we have shown how the model can affect the diversity and complexity of routing. Depending on the specific application, the diversity in turn affects the runtime and quality/cost of the solutions of the corresponding optimization problems. In this section, we describe two case studies demonstrating these impacts of the routing model.

Network Monitoring. When deploying a set of monitoring equipment on a subset of all nodes to observe the status of links, one possible optimization objective might be to use the minimum number of equipment (i.e., minimize deployment *cost*). Finding the right nodes for a deployment is NP-hard in many settings, while for some assumptions efficient exact or approximation algorithms exist [5], [21] for two versions of the problem with one or two types of monitoring equipment.

Note, that this monitoring problem cannot be addressed with tree routing, as not all links of a non-tree graph are used in this case and hence not all links can be monitored. Other routing models can be applied, with varying complexity and cost, provided each link is used in at least one route. In this context, the solution quality refers to the amount of equipment to be deployed: the lower the better.

Traffic Engineering. Consider a graph with capacitated links, i.e., each link has a maximum amount of traffic it can carry. We define congestion to be the ratio between the number of flows using an edge and its capacity. Given a set of requests (flows from v_i to v_j, for $v_i, v_j \in V$), the traffic engineering problem comes in different flavors: assigning routes to each flow, such that either (1) the maxmimum congestion is minimized or that (2) the routes are as short as possible and do not violate the capacity constraints. Both are multicommodity flow problems [10]. The routing model to be used restricts solutions, i.e., the model is expressed as additional constraints in the multicommodity flow problem formulation. In this context, the notion of quality refers to the maximum congestion on a link resp. the length of a capacity-respecting path: lower is better.

A. Runtime

We first discuss the influence on runtime.

Consistency Influences Runtime. One simple example showing that consistency influences complexity regards the traffic engineering of a *single* flow: in a directed network, it is easy and fast to compute a shortest capacity-respecting flow between a given source s and a destination t, e.g., by using Dijkstra's algorithm. However, computing a shortest capacity-respecting route from s to t that fulfills the policy that traffic must go through a single and given waypoint w, is NP-hard [1].

Coherence Influences Runtime. The general unsplittable version of the load minimization problem (i.e., ! model) is NP-complete while optimal routes for arbitrarily splittable flows ($*$ model) can be found in polynomial time [10]. For a variant of the traffic engineering problem, where instead of the concrete traffic matrix upper bounds on the weight of flows from and to nodes are given, a polynomial time algorithm computes an optimal tree routing scheme coinciding with the best possible loop-free confluent routing [12], [14].

We can show similar results when considering the monitoring problem for a graph where the routes are given. We rely on the consistency assumption that the routes use symmetric shortest paths below, regardless of the coherence model applied. It turns out that finding a monitoring deployment is NP-hard for the *any*, *confluent* and *contained* routing model on general graphs (an asymmetric version of this problem with two different types of equipment has been studied in [21], the proofs therein can also be adapted to the symmetric case). For some restricted graph classes, differences in the complexity can be observed. In particular, a polynomial time algorithm can find an optimal assignment for cactus graphs for many routing policies, e.g., for confluent and contained routing (coinciding in this scenario, due to the symmetry), while it is NP-hard even in these graphs under the ! routing model.

Observe that in this case study, the routes are assumed to be given (or chosen by the adversary). In other words, solving the problem only consists in finding a deployment and not in finding a good set of routes as well. The combined routing and monitoring problem is still NP-hard on general graphs, using the same reduction as described above.

B. Quality

Next we study the quality of the optimizations.

Consistency Influences Quality. To study how consistency influences the admissible solution quality, we compare load minimization for different routing models. There exist examples where the minimum congestion achievable with shortest path routing for a given set of commodities exceeds the congestion achievable with other routing models by large factors.

For loop-free routing, Bley [4] shows that the load obtained with shortest path routing can be up to $\Omega(|V|^2)$ times larger than the minimum congestion achievable without this restriction. Furthermore, it is also a factor of $\Omega(|V|)$ larger than the congestion of an optimal confluent routing.

Coherence Influences Quality. For the network monitoring example, we can observe on very simple graphs that the solution quality for different coherence models varies significantly, e.g., for the graph in Figure 4 due to the route diversity. For the model !, we can monitor at most x^2 different routes with shortest path monitoring pairs composed out of x nodes with monitoring equipment. If we can select the routes (i.e., when they are not chosen in an adversarial manner), we thus need at least $\lceil\sqrt{k}/2\rceil$ monitoring equipment for this scenario, deploying half of them on the source nodes on the left and the other half on the destination nodes on the right. For the routing models $>$ and $\subseteq$ the minimum number of equipment we need is linear in k, even when selecting routes is possible.

For traffic engineering, Lorenz et al. [20] show that finding a loop-free minimum congestion confluent $>$ route set is NP-hard. They also show that the minimum congestion may be factor $\Omega(|V|)$ higher for confluent routing than for the any ! routing model. Furthermore it is easy to see that a traffic engineering solution that is restricted to a tree can lead to a solution that is a factor of $\Omega(n)$ worse than a confluent and contained solution, e.g. in a clique with uniform capacities and uniform all-to-all traffic demands.

V. Example: OSPF

Most Internet routing protocols like OSPF and IS-IS conform to the shortest path consistency model, where each link is assigned a weight to represent its cost or length. Furthermore, unsplittable OSPF is confluent and depending on the link weight assignment and/or the selection of one of multiple shortest paths to a destination, OSPF also adheres to the contained routing model. Weights for a contained scheme can be determined efficiently if they exist.

Theorem 3. *(i) Given a loop-free contained scheme, we can find OSPF weights in O(poly(n)) if they exist. (ii) There are loop-free contained schemes for which no OSPF weights exist.*

Proof. (i) Assigning weights to links in a set of routes such that unsplittable OSPF routing can be applied with unique shortest paths is known as the *inverse unique shortest paths problem*. Using linear programming techniques this problem can be solved in polynomial time [3]. With this approach, the ratio between the smallest and highest weight is bounded by the minimum of the number of nodes divided by two and the number of routes in the routing scheme. Any walk including a cycle obviously cannot be described using weights as it is not a shortest path. (ii) Consider the set $S = \{(5,6),(2,3,5),(1,4,5),(4,2,6),(3,1,6)\}$. Since no two paths share two nodes, S constitutes a contained set. However it does not satisfy cyclic compatibility and can hence not be implemented with OSPF [3]. □

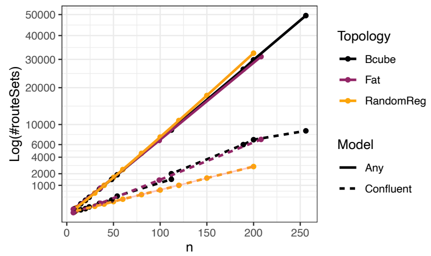

Fig. 5. Number of complete route sets adhering to *shortest paths confluent* and *any* routing models as a function of the topology size, for Fat Trees, BCubes and 3-regular random networks. The ordinate scale is square rooted.

VI. Empirical Model Diversity Analysis

In this section, we take a closer a look at the routing model diversity available in various contemporary network topologies. More precisely, we here focus on shortest path consistent routing, and evaluate the number of *complete* route sets that are !-coherent and $>$-coherent and contain a route for each node pair, denoted by $\kappa^!$ and $\kappa^>$.

As we have seen in the previous section, the choice of a coherence model modifies the solution space of related applications, which in turn impacts both runtime and quality of the obtained solutions. We here empirically compare the size of those solution spaces, namely $\kappa^!$ and $\kappa^>$.

To conduct this numerical evaluation, we focus on 3 types of topologies. First, (LAN) datacenter topologies: we generate all Bcubes and Fat trees of size $n \leq 256$ [1]. Second, (WAN) *zoo* topologies[2], we consider all zoo topologies of size $n \leq 256$. Finally, d-regular random graphs are used to provide a synthetic baseline, with $d = 3$ (average degree of zoo topologies).

Figure 5 plots independently the diversity of both confluent and any for datacenter and zoo topologies. A first observation is the tremendous growth of both values. E.g., for $Fat(4)$ topologies ($n = 99$ nodes), the number of confluent route sets is in the order of 10^{1408} and the number of any route sets in the order of 10^{6740}. This observation holds for all topologies. The straight lines on the square-rooted scale suggest that the number of complete shortest path models is $\approx 10^{n^2}$.

A second observation concerns the fraction of (any) route sets that are confluent, formally the ratio $\kappa^!/\kappa^>$. For instance, on $Fat(4)$, this ratio is 10^{5332} which means that by sampling uniformly any-models, the probability of finding a confluent one is extremely low. This observation is confirmed also for the zoo datasets in Figure 6, which directly plots this ratio. This gap illustrates the considerable scale difference between

[1] We use FNSS for topology generation [**?**].

[2] The Internet Topology Zoo http://www.topology-zoo.org/dataset.html.

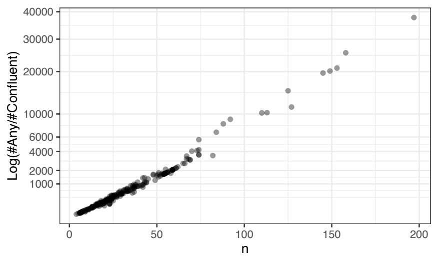

Fig. 6. The ratio between the number of any route sets and *shortest path confluent* route sets on zoo topologies. The ordinate scale is square rooted.

the any route set space, and the confluent route set space. As illustrated before, when optimizing routes for an application goal, this gap impacts both the quality of the optimal route set, and the exploration strategy of the solution space (which in turn impacts the runtime).

VII. Related Work

Only little is known about the impact of the routing model on aspects beyond path diversity. A notable exception is the work by Erlebach et al. [8], [9] who study the computational complexity of routing, under consistency constraints, valid s-t-routes and s-t-cuts, in the valley-free model. Kloeti et al. [17] proposed a generic method applicable to arbitrary graphs for policies which can be formulated as regular expressions (e.g., valley-free, (negative) waypoint routing multipath TCP). Most closely related to our work are [6] and [21]. Chekuri [6] considers the problem of choosing routes for certain demand classes optimally with respect to the resulting congestion and the quality different coherence classes can attain in relation to each other. Chekuri also proposes a hierarchy which we extend in this paper. His *any* (!) property is called *Single Path Routing (SPR)*, his confluent ($>$) property is called *Terminal Tree Routing (TTR)* and *tree routing* (T) has the same name. We extend this hierarchy with *contained* ($\subseteq$), which lies between the confluent and tree model. We also generalize the routing models by allowing to visit nodes multiple times. Pignolet et al. [21] also introduce and compare routing models, however their work is restricted to coherence and does not account for consistency aspects; their results revolve around a specific case study.

VIII. Conclusion and Future Work

This paper was motivated by the observation that the routing model, due to its constraints, can significantly influence the complexity and efficiency at which networks can be operated. We introduced a taxonomy of routing models accordingly and studied different implications. This also led us to question reasoning about network optimization problems in terms of the underlying physical topology only: this graph-centric view ignores the influence of the routing model. Sometimes a more restrictive coherence model does not affect route diversity much while introducing opportunities to speed up algorithms. We also found that tie-breaking among a set of consistent rules can have an impact on the efficiency of some protocols.

We hope that our perspective on consistent and coherent routing models can help the networking community identify novel optimization opportunities and reason about algorithms. In this regard, our paper also opens several interesting avenues for future research. In particular, it remains to explore the impact of the routing model on other performance criteria, e.g., memory/space complexity and on alternative applications, beyond traffic engineering and monitoring.

Acknowledgments. Research supported by the ERC Consolidator project AdjustNet (grant no. 864228).

References

[1] Amiri et al. Charting the algorithmic complexity of waypoint routing. In *Proc. ACM SIGCOMM Computer Communication Review (CCR)*, 2018.
[2] Anderson et al. Netkat: Semantic foundations for networks. In *ACM SIGPLAN Notices*, 2014.
[3] W. Ben-Ameur and E. Gourdin. Internet routing and related topology issues. *SIAM Journal on Discrete Mathematics*, 17(1):18–49, 2003.
[4] A. Bley. Approximability of unsplittable shortest path routing problems. *Networks*, 54(1):23–46, 2009.
[5] P. Boothe, Z. Dvorak, A. Farley, and A. Proskurowski. Graph covering via shortest paths. In *Congressus Numerantium*, 2007.
[6] C. Chekuri. Routing and Network Design with Robustness to Changing or Uncertain Traffic Demands. *SIGACT NEWS*, page 2007, 2007.
[7] M. Chiesa, G. Rétvári, and M. Schapira. Lying your way to better traffic engineering. In *Proc. ACM CONEXT*, 2016.
[8] 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. CAAN*, 2005.
[9] 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.
[10] S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. In *Proc. IEEE FOCS*, 1975.
[11] N. Feamster, J. Rexford, and E. Zegura. The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review (CCR)*, 2014.
[12] J. A. Fingerhut, S. Suri, and J. S. Turner. Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287–309, 1997.
[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
[14] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. In *Proc. ACM STOC*, 2001.
[15] A. J. Gurney and T. G. Griffin. Lexicographic products in metarouting. In *2007 IEEE International Conference on Network Protocols*, pages 113–122. IEEE, 2007.
[16] J. S. Jensen, T. B. Krogh, J. S. Madsen, S. Schmid, J. Srba, and M. T. Thorgersen. P-rex: Fast verification of mpls networks with multiple link failures. In *14th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2018.
[17] R. Kloti, V. Kotronis, B. Ager, and X. Dimitropoulos. Policy-compliant path diversity and bisection bandwidth. In *Proc. IEEE INFOCOM*, 2015.
[18] Kotronis et al. Stitching inter-domain paths over ixps. In *Proc. ACM SOSR*, 2016.
[19] P. Kumar, Y. Yuan, C. Yu, N. Foster, R. Kleinberg, and R. Soulé. Kulfi: Robust traffic engineering using semi-oblivious routing. *arXiv*, 2016.
[20] D. Lorenz, A. Orda, D. Raz, and Y. Shavitt. How good can IP routing be?, 2001. Technical Report 2001-17, DIMACS.
[21] Pignolet et al. Tomographic node placement strategies and the impact of the routing model. In *Proc. ACM SIGMETRICS*, 2018.
[22] G. Rétvári, A. Gulyás, Z. Heszberger, M. Csernai, and J. J. Bíró. Compact policy routing. *Distributed computing*, 26(5-6):309–320, 2013.
[23] J. L. Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 49–60. ACM, 2003.
[24] R. Soulé, S. Basu, R. Kleinberg, E. G. Sirer, and N. Foster. Managing the network with merlin. In *Proc. ACM HotNets*, 2013.