Curvature-based Analysis of Network Connectivity in Private Backbone Infrastructures

LOQMAN SALAMATIAN, Columbia University, USA
SCOTT ANDERSON, University of Wisconsin-Madison, USA
JOSHUA MATTHEWS, University of Wisconsin-Madison, USA
PAUL BARFORD, University of Wisconsin-Madison, USA
WALTER WILLINGER, NIKSUN, Inc., USA
MARK CROVELLA, Boston University, USA

The main premise of this work is that since large cloud providers can and do manipulate probe packets that traverse their privately owned and operated backbones, standard traceroute-based measurement techniques are no longer a reliable means for assessing network connectivity in large cloud provider infrastructures. In response to these developments, we present a new empirical approach for elucidating private connectivity in today's Internet. Our approach relies on using only "light-weight" (*i.e.,* simple, easily-interpretable, and readily available) measurements, but requires applying a "heavy-weight" or advanced mathematical analysis. In particular, we describe a new method for assessing the characteristics of network path connectivity that is based on concepts from Riemannian geometry (*i.e.,* Ricci curvature) and also relies on an array of carefully crafted visualizations (*e.g.,* a novel manifold view of a network's delay space). We demonstrate our method by utilizing latency measurements from RIPE Atlas anchors and virtual machines running in data centers of three large cloud providers to *(i)* study different aspects of connectivity in their private backbones and *(ii)* show how our manifold-based view enables us to expose and visualize critical aspects of this connectivity over different geographic scales.

CCS Concepts: • **Networks** → **Topology analysis and generation**; **Physical topologies**; • **Computing methodologies** → *Learning latent representations*;

ACM Reference Format:
Loqman Salamatian, Scott Anderson, Joshua Matthews, Paul Barford, Walter Willinger, and Mark Crovella. 2022. Curvature-based Analysis of Network Connectivity in Private Backbone Infrastructures. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 1, Article 5 (March 2022), 32 pages. https://doi.org/10.1145/3508025

1 INTRODUCTION

A salient feature of today's Internet is that large cloud and content providers are building and operating their own private global-scale infrastructures (*e.g.,* Google [38, 47, 93], Amazon [17, 73, 94], Microsoft [44, 58], Facebook [9, 48, 75], Akamai [50, 89], Alibaba [1]).[1] These private network

[1] Our focus is on the portion of these companies' infrastructures that is relevant for their role as global-scale cloud provider, not in their role as content provider.

Authors' addresses: Loqman Salamatian, Columbia University , New York, NY, USA, loqman@cs.columbia.edu; Scott Anderson, University of Wisconsin-Madison, Madison, WI, USA; Joshua Matthews, University of Wisconsin-Madison, Madison, WI, USA; Paul Barford, University of Wisconsin-Madison, Madison, WI, USA; Walter Willinger, NIKSUN, Inc. Princeton, NJ, USA; Mark Crovella, Boston University, Boston, MA, USA.

2476-1249/2022/3-ART5 $15.00
https://doi.org/10.1145/3508025

infrastructures serve to minimize the exposure of the traffic generated by their own suite of applications and services to the types of uncertainty (*e.g.,* failures) and variability (*e.g.,* congestion) that their use of the transit provider-based best-effort Internet (*i.e.,* "public" Internet) would entail. Accordingly, increasing portions of the overall Internet traffic utilize these large providers' private network infrastructures and thus bypass the public Internet [84].

The growth and importance of these private infrastructures raise new questions, many of which call for empirical study. Historically, the principal tool used by researchers to understand Internet infrastructure has been `traceroute`. Developed in the late 1980s as a troubleshooting tool for network operators [46], the Internet measurement community has enthusiastically adopted and improved it to study the properties of the routes taken by packets as they traverse the network and to infer connectivity in the underlying physical infrastructure *(e.g.,* [28, 81]).

While many of today's Internet stakeholders continue to support and use `traceroute` for purposes such as debugging, this evolution of the Internet has made the global-scale application of `traceroute` for network measurement and analysis increasingly problematic. More specifically, with the increasing deployment of middleboxes across the Internet [26] and indications that the large cloud providers modify `traceroute` packets that traverse their private backbones (*e.g.,* by manipulating TTL values [39]) or disallow `traceroute` altogether (as documented, for example, in [60] and confirmed by our own measurements efforts detailed in Section 4), the utility of `traceroute` as a widely available, easy-to-use, and reliable technique for (large-scale) Internet measurement experiments may be reaching a point of diminishing return.

In this paper we posit that these trends are inevitable, and that they will continue. Accordingly, we ask the following motivating question: how can we obtain useful insight about network structure when the basic mechanisms relied on by tools such as `traceroute` are either unavailable or can no longer be taken for granted?

In considering how to answer this question, we argue that the increasing opacity of path internals in today's Internet calls for leveraging measurement techniques that are both more light-weight and more universally-available than `traceroute`, combined with more heavy-weight mathematical analysis tools for extracting as much information as possible from these more limited information sources. To this end, we consider the *end-to-end round trip delay* (RTT) along an Internet path as an information source that is suitably light-weight and at the same time essentially universally-available. Moreover, RTT can in most cases of interest also be readily augmented with metadata in the form of approximate geolocation of path endpoints. While neither RTT nor geolocation can be measured with high precision for every path in the Internet, we find that in general, the necessary information is available with sufficient precision. At the same time, our analysis of the resulting measurements is informed by mathematical concepts from Riemannian geometry (*e.g.,* Ricci curvature and its extension to discrete graphs known as Ollivier-Ricci curvature [66]) and presented in the form of specialized visualizations (*e.g.,* continuous manifolds with embedded geo-information). This considered heavy-weight analysis tool only assumes that the "footprint" of a given network's physical infrastructure is specified in terms of a set of geo-located nodes that represent vantage points capable of performing the light-weight measurements of interest (*e.g.,* RTT).

Thus, as an answer to our motivating question, we show in this paper that in a "post-`traceroute`" world, it is possible to rely solely on RTT measurements among a set of geo-located Internet nodes to elucidate important aspect of Internet structure. Specifically, we assemble a set of existing and newly-developed techniques into a coherent and original methodology for illuminating important aspects of network structure – aspects that are either impossible or more difficult for `traceroute` to identify and discover as its ability to discern path internals in the Internet diminishes.

We demonstrate our methodology by applying it to three large cloud provider networks (Google, Microsoft, and Amazon). Specifically, we use inter-data center RTT measurements from each network to identify critical aspects of connectivity within each infrastructure to compare and contrast across infrastructures. These results show that our approach enables third-party Internet measurement researchers to gain insight into the private backbone infrastructures of the large cloud/content providers, which is critical for independently verifying claims by these providers about the characteristics and performance of their network. Importantly, our approach does not rely on information concerning their physical connectivity fabrics which these providers increasingly view as proprietary in nature, mainly for competitive reasons.

Contributions: In summary, in this paper we make the following contributions. First, we present a new methodology for assessing connectivity in private backbone infrastructures. By leveraging simple RTT measurements among a set of geo-located nodes and employing concepts of Riemannian geometry as a mathematical vehicle, our approach offers a promising alternative to inferring important aspects of network structure in a post-`traceroute` world. A key element of our method is a coordinated array of interrelated visualizations, including a new visualization tool developed for this work that fuses Riemannian geometry with geographical maps. Second, we demonstrate our methodology by inferring known and novel aspects of existing connectivity fabrics of the private backbones of the large cloud providers. For example, by comparing the inferred connectivity fabrics of three large cloud providers, we are able to highlight common features and important differences and identify "weak spots" in their existing connectivity. Based on these results, we argue that our proposed approach presents an exciting alternative to traditional methods for Internet connectivity research and has the potential to inform relevant stakeholders on what future infrastructure changes would be most beneficial (or detrimental) for some large cloud providers. Such guidance is of critical importance for an industry that is expected to spend up to $150 billion on new fiber deployments in the U.S. alone [71].
This work raises no ethical considerations.

2 BACKGROUND

The Problem. Since the early 1990s, the preferred method for studying Internet infrastructure has been to infer router-level topologies based on information gleaned from sufficiently many `traceroute` measurements. The key idea behind `traceroute` is to elicit a self-identifying response from *intermediate* nodes along an end-to-end path. While many ingenious strategies have been developed for making this idea maximally informative [15] and efficient [18], much can nonetheless go wrong when performing this inference task in practice (*e.g.,* see [61] and references therein).

First of all, since the very beginning, `tracereoute`-based measurement studies have faced an array of technical challenges and barriers: *(i)* routers that emit their response from a randomly chosen interface; *(ii)* routers that do not respond at all to TTL exceeded triggers or ICMP probes; *(iii)* ambiguities in mapping interfaces to routers (IP alias resolution problem); and *(iv)* bias due to choice of vantage points for launching `traceroute` probes.

Next, the increasing complexity of the Internet is making `traceroute` less effective. The basic mechanism used by `traceroute` is unable to detect the use of underlying layer-2 technologies (*e.g.,* MPLS), and in the presence of those technologies, `traceroute` returns misleading results [80]. Further, the Internet has seen a rise in deployed middleboxes such as firewalls, NATs, proxies, and Deep Packet Inspection boxes. These middleboxes are used to manipulate traffic for purposes other than simple packet forwarding (*e.g.,* for performance or security reasons), and their ability to drop packets that are carrying probe information can render traceroute as a means for exploring a network's infrastructure largely useless.[2]

Finally, the global Internet ecosystem itself is changing in ways that make traceroute less effective. The well-documented flattening of the Internet is the result of many of the large multinational technology companies building out private global-scale infrastructures that often carry traffic from the origin all the way to the access network. These increasingly dominant infrastructures are opaque to traceroute. Further, global-scale cloud providers that serve as hosts for third parties are equally problematic. For one, to be able to perform relevant measurements, third party researchers have to become customers of such a provider in the sense that they have to purchase resources from the provider in the form of virtual machines (VMs). Moreover, to perform the desired measurements, researchers must adhere to the terms and conditions imposed by the provider that often prevent them from running tools such traceroute at will. And, even if performing measurement experiments such as running massive traceroute campaigns is technically permitted, it is completely up to each provider to determine how traceroute probes are handled within their own domain (*e.g.,* see discussion in [13] and references therein).

The result of all these challenges and trends is that currently inferred router-level topologies are of largely unknown quality (*e.g.,* different degrees of completeness and/or accuracy; see for example [81]). In some cases, inferred connectivity fabrics have nothing to do with router-level connectivity but instead provide instances of logical connectivity, or worse, present an entangled and largely meaningless mixture of layer-3 and layer-2 connectivity (*e.g.,* see [54, 88]). Furthermore, technology trends suggest that these problems will not go away, leading us to conclude that the practical utility of traceroute for studying and characterizing Internet infrastructure can no longer be taken for granted and requires continued careful scrutiny. In particular, this observation argues for renewed efforts to develop alternative Internet measurement tools that are *(i)* easily deployable, *(ii)* generate probe packets that are not susceptible to tempering by third parties, and *(iii)* yield high quality end-to-end information that is comparable to a properly-executed traceroute probe.

What can be done? When one considers the sorts of measurements that necessarily must be available to a third party researcher based on the Internet's dominant service model, it becomes clear that only end-to-end measurements are guaranteed to be available. Of these, the simplest is end-to-end round trip time (RTT) or delay. While simple ICMP ECHO probes can be blocked, RTT can be obtained through TCP connections (SYN/SYN-ACK) for any host that has an open port. Furthermore, it is generally possible to obtain at least approximate geo-location information about end systems. For example, while with RTT measurements, delay-based geo-location is always possible, in many cases of practical interest, more precise geo-location information is available. As a result, we assume that neither RTT nor geolocation information can fundamentally be completely hidden by network operators from third-party researchers. Although both data sources can be manipulated, the capabilities of an adversary for doing so are limited [37].

Starting from these two data sources, we envision a characterization of network structure in terms of the properties of paths between end systems. Our approach is to assume that the data (*e.g.,* measured network latencies) has been generated from an unknown complex topological space (*e.g.,* continuous manifold), and our methodology provides a means for "learning" this manifold; that is, uncovering the geometry of this unknown space. In particular, the data points represent shortest paths (along the manifold's surface between measurement nodes) that encode a composite view of underlying physical network infrastructure and deployed routing configurations. Intuitively,

[2] While extensions to traceroute such as tracebox [26] can help with identifying various types of middlebox-related packet modifications and pinpointing where on a path a given middlebox caused packets to be modified, they do not eliminate the basic problem.

the “curved" nature of the manifold sheds light on important local properties of the network, with positively curved regions representing more or less pronounced peaks (resulting in shortest paths that are indeed “direct") and negatively curved areas forming saddle point-like shapes (producing shortest paths that are “long") [64, 67]. We expand on the connection between our approach and continuous manifolds in Section 3. While our proposed method is not designed to provide the same amount of information as `traceroute` and is therefore not intended as a replacement of `traceroute`, we illustrate that it enables us to obtain insights into network connectivity even in situations where `traceroute` probes are blocked or manipulated or result in measurements that are difficult to interpret.

Our methodology enables both local and global characterizations of a network’s infrastructure. At the local level, our method assigns two properties to each pair of nodes: *importance* and *performance*. Importance is a local measure of connectivity (*i.e.,* “curvature"). It identifies node pairs whose connections play a critical role in defining the structure of the network. Performance is a local measure of the optimality (or lack thereof) of the path connecting the nodes. It identifies node pairs whose connectivity could conceivably be improved. On the other hand, at the global level, our method provides a macroscopic view of network nodes as forming well-performing “clusters" over a range of performance levels. By sweeping through a range of performance levels, a picture emerges that characterizes portions of a network in which performance is good, and portions between which performance could be improved.

3 METHODS

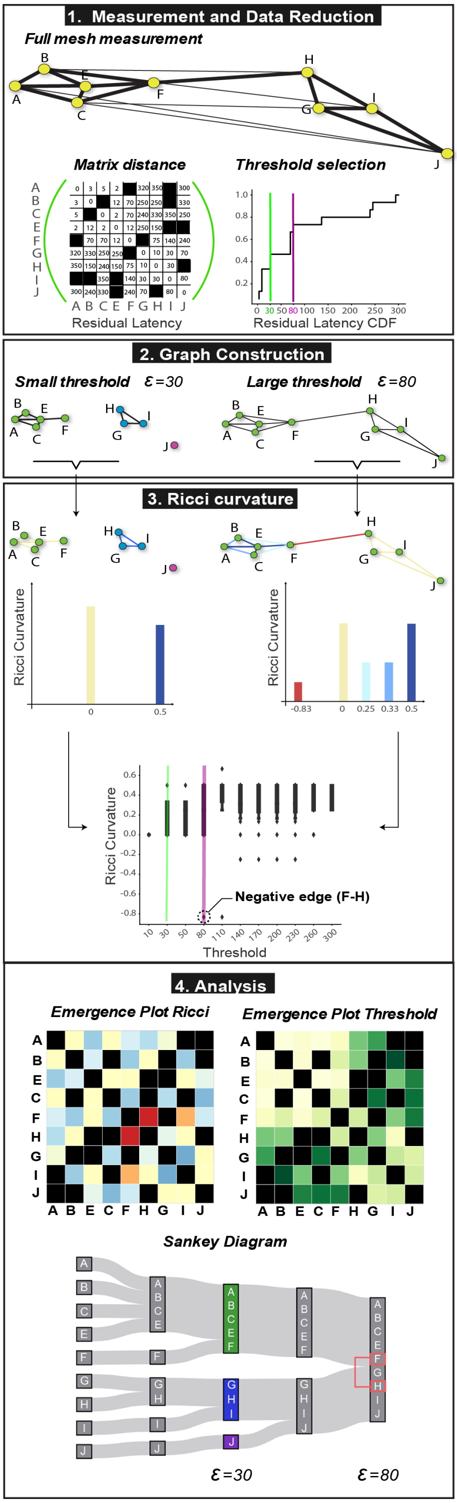
Overview. A schematic overview of our method is presented in Figure 1. In what follows, we explain the steps shown in the figure.

The system under study is a physical infrastructure connecting a given set of nodes (vantage points). In general, the physical connectivity fabric (shown as thick black lines in the figure’s topmost panel) is unknown. We seek to develop methods that elucidate connectivity within the system that are very lightweight and thus well-suited to large-scale assessment in settings where measurement capabilities may be limited (*e.g.,* cloud infrastructures). To achieve this, we depend on only two simple information sources: *(a)* geographic location of nodes, and *(b)* round trip latency between node pairs.[3]

Hence, the starting point of our method, shown in the topmost panel of the figure (*i.e.,* panel 1), is the collection of a mesh of all-pairs latency measurements among a set of nodes, but as indicated in panel 1, we allow for a small portion of missing measurements. Note that the more localized the target area of study, the more precise the latency measurements should be for our methodology to apply. For each pair of nodes with latency measurements, we determine the *great circle latency*, which we define to be the great circle distance (GCD) between the nodes divided by $\frac{2}{3}c$, which is the speed of light through fiber [19]. Then, for each node pair, we compute the *residual latency*, which we define to be the measured latency minus the great circle latency. Residual latency can be influenced by a number of factors including physical paths, logical connectivity defined by routing and congestion. Our focus on private infrastructure and minRTTs obviates the latter two issues, which we will consider in the context of the public Internet in future work (see Section 8). The resulting matrix X of residual latencies is shown on the left of panel 1 in the figure, with black squares indicating missing observations.

The next step is to identify node pairs that are connected by *nearly straight* paths. By ‘straight’ we mean following a great circle (geodesic), and by ‘nearly’ we mean that the total deviation from the great circle is bounded. We formalize this notion by establishing a threshold ε on residual

[3] When we refer to *measured latency* or *latency* in this section, we actually mean minimum RTT (minRTT).



	A	B	C	E	F	G	H	I	J
A	0	3	5	2	■	320	350	■	300
B	3	0	■	12	70	250	250	■	330
C	5	■	0	2	70	240	330	350	250
E	2	12	2	0	12	250	150	■	■
F	■	70	70	12	0	■	75	140	240
G	320	330	250	250	■	0	10	30	70
H	350	150	240	150	75	10	0	30	■
I	■	■	350	■	140	30	30	0	80
J	300	240	330	■	240	70	■	80	0

Fig. 1. Our methodology in a nutshell.

latencies. Because it measures how close a measured latency is to its theoretical optimal, we call ε the *performance threshold.* This step appears on the right side of panel 1 in the figure, where the CDF of residual latencies is shown along with two different thresholds.

For any given ε, the set of node pairs that are connected via 'nearly straight' paths defines a graph. This graph is defined as the set of edges that have residual latency $\leq \varepsilon$ and shown in panel 2 of the figure for two different values of the performance threshold ε. A lower threshold defines a graph with fewer edges; edges in the left hand graph are a subset of the edges in the right hand graph. The low-threshold graph reveals structure locally that allows us to infer the underlying physical connectivity fabric of the various disconnected components. However, it says nothing about how these islands are connected physically as part of the overall graph. That information can be inferred from what higher-threshold scenarios reveal in terms of important links that are on many shortest paths between nodes in the different local 'regions' of the overall infrastructure.

The next step is to assess the *importance* of each edge in the graph. For that purpose, we use the notion of Ricci curvature of an edge. We discuss curvature and how it is computed below. Here we offer the following heuristic view of curvature: negatively curved edges will tend to be traversed by many of the local shortest paths; that is, paths that originate in the neighborhood of one end node of a negatively curved edge and connect to nodes in the neighborhood of the node at the other end of that edge. Conversely, edges that are positively curved will by and large not be used by most of the local shortest paths. The resulting graphs in which each edge is assigned its curvature (shown by color) are depicted at the top of the third panel of the figure.

Next, to obtain a macroscopic view of each graph, we compute the distribution of edge curvatures present in the graph. This allows us to understand the relative proportion of curvatures among graph edges. These distributions are shown in the middle of panel 3 of the figure. Finally, we obtain a thorough characterization of each dataset by repeating the above procedure over a range of many performance thresholds. By doing so, we obtain a picture of how the graphs derived from our data evolve as we relax the constraint implied by using 'nearly straight' paths. The distributions of resulting curvature values are summarized in a series of boxplots as shown on the bottom row of panel 3 in the figure.

The figure illustrates a number of aspects of our methodology. It shows that at small thresholds, we typically obtain a disconnected graph. At this level, only node pairs with very 'straight' paths are connected. As the threshold increases, additional edges are added, and at some point the graph becomes connected. The figure also shows that edges that appear to be 'bottlenecks' or 'bridges' tend to show negative curvature. Such edges indicate critical (important) connectivity; removing such edges tends to disconnect the graph (or at least drastically lengthen the graph's local shortest paths).

Putting it all together. Panel 4 of Figure 1 shows how we synthesize the above metrics into a complete picture. To do so, we look at the behavior of both nodes and edges. With respect to nodes, we characterize how nodes cluster as the performance threshold varies. To capture the entire picture of node clustering at different performance thresholds, we use a *Sankey diagram* as shown at the bottom of panel 4 in the figure. This plot gives a macroscopic view of how the threshold-dependent curvature-weighted graphs evolve as the performance threshold increases from left to right. In particular, it shows which nodes form well-performing clusters at low performance thresholds (left side of the Sankey diagram), which nodes form larger connected components at higher performance thresholds (right side of Sankey diagram), and how the different clusters merge or split as we vary the performance threshold from low to high or vice versa.

Augmenting this node-based view, we also characterize how edges contribute to connectivity of the overall network. To do so, we examine the properties of an edge at the threshold where the edge first emerges (*i.e.,* the lowest $\varepsilon \geq$ the edge's residual latency). We define two metrics for each edge, namely *importance* and *performance*. We consider an edge's importance as its impact on network robustness – *e.g.,* how much 'disruption' (*i.e.,* affecting locally shortest paths) would occur were this edge not present. For this, we use Ricci curvature as our metric, with lower (more negative) Ricci curvature indicative of greater edge importance. For edge performance, we use its residual latency – *e.g.,* the value of ε at which the edge appears in the graph. Here, higher values indicate poorer-performing edges, *i.e.,* edges whose latencies are more inflated with respect to the optimal.

We summarize the two edge metrics in the form of heatmaps as shown in panel 4 of the figure. For edge importance (Ricci curvature), key network edges appear as very negative (red in the figure). For edge performance, well-performing edges (low residual latency) appear as having the lowest values (white or yellow in the figure). In each figure we use black boxes where no per-edge measurements are available.

Ricci Curvature. A key tool in our approach is the *Ollivier-Ricci curvature* [66].[4] Roughly, the Ricci curvature of a graph edge can be thought of as a sort of *local* 'betweenness' measure. However, it has important differences from betweenness. Rather than counting paths (as in betweenness), it measures the *optimal transport* of mass from the neighbor of one edge node to the neighbor of the other edge node. Hence we first discuss optimal transport, and then we use it to define Ollivier-Ricci curvature.

Optimal Transport. To formally define the optimal transport problem, we consider a set of m sources that produce a commodity, with source i producing amount x_i of the commodity. We also consider a set of n sinks, each wanting to consume an amount y_j of the commodity. The cost of transferring one unit of the commodity from source i to sink j is given by a distance metric $d(i, j)$. The optimal transport problem finds a *transport plan* describing for each (i, j), how much of the commodity produced at source i should be transported to sink j, so as to minimize the total cost of transport, while obeying commodity conservation constraints. Depending on specific assumptions, such as infinite divisibility of the commodity, the problem may be solvable via a linear program, which is the case in our work.

More formally, given a metric space (M, d) meeting certain conditions, the 1-Wasserstein distance (also referred to as 'earth mover's distance') between two probability measures μ and ν on M is defined as

$$W_1(\mu,\nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \int_{M\times M} d(x,y)\,\mathrm{d}\gamma(x,y)$$

where $\Gamma(\mu,\nu)$ denotes the collection of all measures on $M \times M$ with marginals μ and ν on the first and second factors, respectively. The intuition is that γ describes a plan for moving a mass from distribution μ to distribution ν, and the amount moved from x to y (*i.e.,* $\mathrm{d}\gamma(x,y)$) has unit cost $d(x,y)$.

Ollivier-Ricci Curvature. Using the concept of optimal transport, we can define the Ollivier-Ricci curvature of an edge connecting nodes x and y. We start by dividing and placing a unit mass equally over the neighbors of x, denoted by $N(x)$. We also divide another unit mass among the neighbors of y, denoted $N(y)$. These two unit masses define the distributions μ and ν. We measure distance between any two nodes in the graph using hop distance. With this setup of sources $N(x)$ and sinks $N(y)$, we compute the optimal transport cost of moving the unit mass from the neighbors of x (*i.e.,* μ) to the neighbors of y (*i.e.,* ν). The Ollivier-Ricci curvature of the edge $x \sim y$ is then given by

$$\kappa(x,y) = 1 - \frac{W_1(\mu,\nu)}{d(x,y)}.$$

[4] For brevity we will usually write just 'Ricci curvature.'

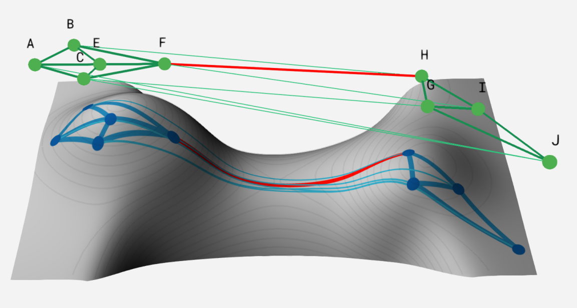


Fig. 2. A manifold view of the example network used in Figure 1.

A few examples can help illustrate the main idea. Consider first an edge between x and y that is part of a tree. Then the neighbors of x (*i.e.,* $N(x)$) and y (*i.e.,* $N(y)$) are fully disconnected and all the mass from $N(x)$ must go through the $x \sim y$ edge in order to get to $N(y)$. The optimal transport cost of moving $N(x)$ to $N(y)$ is equal to $d(N(x), x) + d(x, y) + d(N(y), y) = 3$ and we see that the edge is negatively curved with a value of -2. Next, consider a situation in which x and y have the same number of neighbors, and each neighbor of x has a direct connection to the corresponding neighbor of y, approximately like a grid. Then all mass moves through hops of length 1 and the edge has zero curvature (it is 'flat'). Finally, consider a clique. In this case, $N(x) = N(y)$ and no mass moves at all – so the curvature is positive (+1).

These examples illustrate why we use Ricci curvature to characterize an edge's importance. Negatively curved edges – those through which much mass must pass – represent local bottlenecks in the graph. A change to the performance of one of these edges can have a significant impact on the performance of many paths in the network.

The manifold view. The graphs we construct represent paths between locations in geographical space. Our results show that it is helpful to understand the relationship between the curvature of edges in our graphs and the curvature of the underlying geographical space. However, while the notion of curvature is more intuitive for continuous (Riemannian) surfaces than it is for graphs, the problem of representing combinatorial objects such as graphs using smooth topological spaces such as manifolds is notoriously difficult. We address this problem by appealing to a well-known relationship between the Ollivier-Ricci curvature and the standard Ricci curvature (for more details, see for example [52, 66]).

A key visualization component of our method is a representation we refer to as the *manifold view.*. Informally, to generate this manifold view, we start from a "flat" (2-D) geographical map and a geographically embedded graph (with associated edge curvatures) and then produce a new (3-D) surface whose local Gaussian curvature in each region approximates the Ricci curvature of the edges that "pass near" that region. Figure 2 illustrates the concept, using the example network from Figure 1. The figure shows that the positively curved edges in the graph (*e.g.,* edges connecting nodes A-F, or nodes G-J) induce two positively curved peaks in the manifold view, and the negatively curved edge (*e.g.,* F-H) induces a saddle shape in the manifold view.

While a more formal derivation of this manifold view is beyond the scope of this paper and will be presented in future work, we briefly describe in the following the basic steps required to implement our novel visualization approach. Since adequately accounting for the positive and negative curvature features in complex manifolds is a known and challenging problem, we expect future versions of our basic tool to result in improved representations of our proposed manifold view.

We start from a geographic map represented as a discrete mesh of points $\{\mathbf{s}_i = (x_i, y_i, z_i)\}, i = 1, \ldots, N$. In the mesh, each node is connected to its neighbors to form a triangulation of the surface we seek to generate. The discrete approximation to Gaussian curvature at each vertex $\mathbf{s}_i$ is given by

$$c(i) = 2\pi - \sum \text{angles of the mesh around vertex } \mathbf{s}_i.$$

We assign each vertex of the mesh a target curvature $\kappa(i)$ based on the average curvature of the graph edges that pass close to the vertex, via a kernel smoothing function. We then define an objective function:

$$\mathcal{L}(\mathbf{z}) = \sum_i (\kappa(i) - c(i))^2 + \lambda \mathbf{z}^T L^3 \mathbf{z} \tag{1}$$

where $\mathbf{z} = [z_i]^T$. The first term in (1) ensures that the curvature of the mesh $c(i)$ is close to the desired curvature $\kappa(i)$ at each mesh vertex i. The second term in (1) enforces a smoothness constraint on the second derivative of the surface, using L, the graph Laplacian of the mesh, as an approximation for the Laplace-Beltrami operator on the surface. The parameter λ determines the smoothness of the mesh curvature and is set manually based on visual inspection. We use a fixed value of λ across all visualizations to enable cross-dataset comparisons. Keeping the x and y coordinates of the mesh fixed, our method finds the best z (elevation) for each point on the surface via gradient descent applied to the objective function $\mathcal{L}(\mathbf{z})$.

The figure illustrates the key ideas of the manifold view. A negative curvature of an edge in the graph arises when local paths in the graph tend to make heavy use of that edge. In a sense, shortest-paths (geodesics) are "attracted" to the edge, and may be thus "lengthened," by local connectivity properties. Negative curvature in the manifold induces the saddle-shape seen in the center of the figure. Similarly, geodesics on the manifold will tend to curve inward towards each other, which attracts the paths towards each other and lengthens them (compared to a flat surface). Analogous statements hold for positive curvature in the graph and on the manifold. Thus the manifold view gives a visual sense of how paths in a geographical setting will tend to be elongated or shortened by the underlying connectivity of the embedded graph.

Discussion. The manifold views of networks generated by our methods should not be thought of as direct replacements for connectivity maps generated from `traceroute` measurements *e.g.*, [81]). They are unique representations that convey key characteristics of networks related to connectivity, optimality and performance. Our graphs and manifolds relate to physical connectivity in terms of the geolocations of nodes and geodesic distances and aspects of logical connectivity in terms of RTT measurements between nodes.

While we have already described the limitations of `traceroute`, an alternative is maps of physical network connectivity that may be published by providers (*e.g.*, [59]). While these can be useful in certain contexts, they can be incomplete [28], out of date, and difficult to interpret. We posit that the views generated by our methods provide unique information when maps are not available and complementary information when maps are available.

4 SENSITIVITY ANALYSIS

In developing our methodology, we make a number of simplifying assumptions. In this section, we discuss our choice of using residual latencies that are defined as the *difference* rather than as the *quotient* of the observed and the GCD-defined latencies and explain our reasons for using Ricci curvature as our metric-of-choice instead of other more standard measures of centrality.

Residual latency in absolute vs relative terms: The main reason we selected *difference* over *quotient* for residual latency is that it allows us to perform a gradual and geographically meaningful discovery of the underlying metric space. When residual latency is defined in terms of *quotients*, we found that the graph percolates in an unintuitive fashion. In particular, the first edges that emerge are “mid-range" edges within different continents. Many nodes that are geographically close to each other exhibit poor performance due to last-mile congestion and metro area approximations, which in turn implies that they appear to have high residual latency. This makes the interpretation of the results more difficult, especially in terms of the manifold view, where we expect almost Euclidean behavior for local areas. Another key contrast is that residual latency based on *difference* induces a unit while the ratio is a relative measure. In summary, while *quotient* is a viable choice for defining residual latency and for comparing networks (*e.g.,* for performing a similar analysis as in Section 6), our intuition and representations are built on the premises that *(i)* the 2D Euclidean projection manifold should encapsulate the expected behavior of straight paths and *(ii)* the curvature models the distortion from the projection to the actual metric space that the network creates. In that context, the difference between the flat plane and curved Internet yields a more tractable way of discovering the curvature.

Using Ricci curvature as centrality metric-of-choice. Edge betweeness [25, 70] and the cut metric [20] are two commonly-used metrics for characterizing edge importance in a graph. Compared to these metrics, using the Ricci curvature metric to measure edge importance has a number of advantages. First, the Ricci curvature is able to reveal the same features that edge betweenness and the cut metric can identify. Second, by opting for the Ricci curvature, a key strength of our method is its ability to focus on a given graph edge and track the evolution of the curvature of that edge as the graph changes in response to the selection of different threshold values. Neither edge betweeness nor the cut metric have this desirable property. In fact, in Appendix C, we show that both metrics can be highly unstable and sensitive with respect to the addition or deletion of certain edges and are therefore ill-suited for tracking such evolutionary aspects of graphs. Third, the Ricci curvature provides a means for a geometric interpretation of graph-related properties which we heavily rely on when generating our manifolds.

5 DATA

The empirical component of our work focuses primarily on collecting round trip time (latency) measurements on network paths between data centers for each of the major cloud providers (*i.e.,* traversing links owned and operated by each cloud provider). Our specific objective is to find the minRTT from all measurements collected over a period of days between widely deployed measurement points. We note that since our approach described in Section 3 will generate the same results from minRTT and $\frac{1}{2}\times$ minRTT, we use minRTT in our study because minRTT is directly measured without additional assumptions about the specific network path taken. While it is possible to study minRTT over fixed-length periods that are either shorter or longer than a few days, such investigations are beyond the scope of this work. For example, using our method with minRTT values that have been computed over shorter time periods (*e.g.,* minutes or hours) can be expected to shed light on changes in overall network performance such as diurnal patterns or unusual behavior. On the other hand, analyzing minRTT values that have been computed over time intervals on the order of weeks or months can provide valuable information for network operators concerned with network capacity planning in light of observed long-term trends in measured end-to-end delays. As part of our future work, we are particularly interested in investigating how our methodology can be leveraged to study more real-time network conditions by means of generating and analyzing a set of evolving graphs where each individual graph encodes critical network performance-specific information that has been measured over short periods of time. Also, since our goal is a representative measure of latency along a path, we limit our measurements to intra-cloud provider networks to reduce the variability from route dynamics and path asymmetries, understanding that both affect RTT measurements. Finally, we only consider paths and connectivity available via IPv4, leaving IPv6 studies to future work.

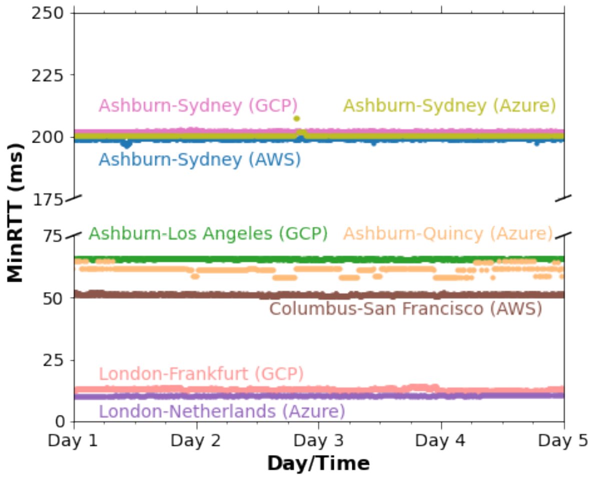


Fig. 3. Minimum round trip time measurements over a 5 day period for different node pairs connected via private cloud networks.

Our latency measurements for private Internet paths focus on Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). We collected measurements between each city pair of datacenters for the above-mentioned cloud providers. While we did not have control over the specific datacenter in a given cloud location, we argue that control at this level is not material to our findings because it only results in differences in latency that are on the order of microseconds. We approximated the physical location between data centers in a city as the city center. We use RIPE Atlas anchors and probes to collect RTT measurements for AWS. Since Google and Azure did not host RIPE anchors in their data centers at the time when the measurement campaigns were initiated,[5] we configured virtual machines on servers in their data centers around the world and used *ping* to collect RTT measurements in a full mesh between sites. Details of data center locations are provided in Appendix B.

Using virtual machines for measurement, as opposed to running measurement code from an OS installed directly on the hardware, does incur an additional performance penalty of <0.5 ms for VMs running on KVM or XEN hypervisors [35, 85, 92]. We acknowledge this overhead, but do not see it as a significant source of error as our minimum minRTT measurement was ~5.5 ms. In addition to the VM latency overhead, [77, 86] noted much variation between individual measurements and much delayed initial ping RTT. To account for these performance penalties, we sent multiple ping probes during each measurement period, took measurements over multiple days, and only considered the minRTT from all measurements for our curvature analysis.

AWS: We computed RTT measurements from RIPE Anchors (and RIPE Probes for 4 locations) in 21 AWS locations between 12/18/2019 and 12/23/2019. Regular measurements between Atlas anchors include 6 standard ping measurements taken at four minute intervals and 6 standard `traceroute` measurements at 15 minute intervals. We mapped the observed `traceroutes` to AS paths using standard prefix mapping and conclude that the traffic remains within the boundary of AWS network.

Azure: We collected RTT measurements from 17 Azure cloud locations (*i.e.*, availability zones) from 12/21/2019 to 12/25/2019. Each virtual machine in an Azure data center had at least one vCPU, 1GB RAM, 4GB of storage, and all ran Ubuntu Server 18.04 LTS. We configured network peering and firewall rules to allow communication between VMs in the various locations. Each VM was

[5] Google has since started hosting RIPE Atlas Anchors in most of its locations.

assigned a public IP address and a private IP address. We collected RTT measurements using both the public and the private IP addresses. Each VM sent 20 ping (ICMP echo) requests to each of the other VMs every fifteen minutes.

In contrast to the other cloud providers, on Azure the default `traceroute` application does not produce results. Instead of `traceroute` we used `nmap` with a SYN scan to port 80 and the `traceroute` flag. With this configuration, `traceroutes` between VMs with private IP addresses indicated a direct connection and `traceroutes` between VMs with public IP addresses indicated multiple intermediate hops that did not acknowledge the packet; the only intermediate hops that did respond were from addresses owned by Microsoft.

Google: We measured RTT between 24 GCP data centers from 11/08/2020 to 11/14/2020. Each virtual machine in a Google VM is a g1-small (vCPU: shared, RAM: 1.7 GB), running Ubuntu 18.04 and with a static IP address. For each VM, we collect ping measurements to the other 23 GCP VMs once every 4 minutes (matching the RIPE Atlas probes). For each ping measurement, we use the standard ping utility to send 10 probes at an interval of 0.2 sec and only consider the minRTT.

We collected `traceroute` measurements between VMs using private IP addresses. The measurements indicated that the VMs were directly connected. We also configured public IP addresses for each VM using the GCP standard network tier *i.e.,* hot-potato routing for egress traffic and akin behavior for ingress traffic. Although multiple hops along the path between VMs responded to the `traceroute` probes, prefix mapping the responding IP addresses indicated that all traffic remained within the Google WAN. When we assigned public IP addresses to VMs using the GCP premium network tier *i.e.,* cold-potato routing [4], the characteristics of responses to `traceroute` between VMs hosted on the GCP infrastructure did not change. However, `traceroute` to and from RIPE Atlas anchors outside of GCP indicated that the probes entered the GCP WAN as close to the originating source as possible.

Figure 3 provides examples of the minRTT measurements for each collection interval between datacenter locations that are connected via private paths. The location pairs in the figure were selected to highlight relatively short, medium and long geographic paths that are considered in our study. There is relatively little variation in minRTT over the measurement period. The figure also shows clear evidence of the use of traffic engineering, as indicated by the different but consistent minRTT values across many of the paths in their infrastructures.

6 THROUGH THE LENS OF RICCI CURVATURE: CLOUD PROVIDER NETWORKS

Many of the large cloud/content providers are busy expanding their private backbone networks, but at the same time they often view detailed information about their physical infrastructure to be proprietary and have become reticent to share the details publicly. These developments beg the question: as access to data about the physical infrastructures of the backbones of the large cloud/content providers is harder to come by while at the same time the importance of these private networks increases, is it possible to study the pertinent characteristics of these private backbones without having detailed knowledge of their connectivity fabrics (*i.e.,* existence and/or locations of fiber optic cable conduits)? We answer this question in the affirmative by applying our methodology to examine the critical features of three private global-scale backbone networks: Microsoft Azure (Azure), Amazon Web Services (AWS) and Google Cloud (GC). As noted above, we scheduled measurements in each of the data centers that these providers operate in their respective availability zones across the globe. Availability zones can be associated with the physical nodes of these providers' global-scale private backbones whose connectivity fabrics we attempt to elucidate. By performing all-pairs latency measurements among these sets of nodes and annotating each (logical) edge between them with the corresponding measured minRTT value, we obtain the analogue of the complete graph shown in Panel 1 in Figure 1 for each provider. The main results of our curvature-based analysis of private backbone networks are discussed below.[6]

6.1 Microsoft Azure as a case study

We conducted a case study of Microsoft Azure based on data collected from virtual machines in each of the 17 data centers that this provider operates across the globe. As described in Section 5, we performed all-pairs latency measurements among this set of nodes and annotated each (logical) edge between them with the corresponding measured minRTT value. Using GCD as the measure for physical path distance, we apply our methodology as described in Section 3. The main results of our curvature-based analysis of Azure's private backbone network are shown in the Azure column in Figure 4.

Panel (a) in the Azure column of Figure 4 shows that there is a broad spectrum of performance thresholds, mainly due the inclusion of submarine cables that result in residual latencies that tend to be significantly larger than those encountered in a strictly intra-continental setting. The green-colored threshold value of 60 in panel (a) results in the first edges with negative curvature. Azure's connectivity structure at threshold 60 is shown in panel (c) and identifies the existence of two pronounced and geographically meaningful connected components: a North America-Europe cluster and an Asia-Pacific cluster. Within the North America-Europe cluster, all intra-North America edges have positive curvature as do all intra-Europe edges – an indication that Azure operates a richly connected physical infrastructure within both the North American and the European continents. The formation of this North America-Europe cluster is due to three edges that connect Virginia, US and the UK-Ireland. Consulting the Azure global network map [59] confirms the existence of direct under-sea cables connecting Ashburn, VA and London, and Boydton and Dublin, respectively. In contrast to the North America-Europe cluster, the Asia-Pacific cluster has a more unbalanced internal connectivity. In particular, while Mumbai is only connected to Singapore, there exists rich connectivity among the other four intra-cluster cities Singapore, Hong Kong, Seoul, and Tokyo. This intra-cluster connectivity suggests the existence of an undersea cable between Mumbai and Singapore, but the fact that this edge's curvature is not negative illustrates that Mumbai does not play the role of a gateway node and that this edge is not critical for connectivity within the Asia-Pacific cluster.

Analyzing the graph in panel (d) that corresponds to threshold 90 colored in purple in panel (a), we see that the entire North America-Europe cluster now consists exclusively of positively curved edges, which implies the existence of rich fiber connectivity linking the two continents by means of transatlantic cables with a diverse set of landing points. At the same time, examining the Asia-Pacific cluster at this threshold, we find that while it now includes Sydney, its intra-cluster connection fabric is also positively curved but far more sparse than that of the North America-Europe cluster. Importantly, we observe the presence of a single negatively curved edge between Tokyo and Quincy, WA (shown in red) that connects the North America-Europe and Asia-Pacific clusters.

Finally, in panel (e) we see that at threshold 120, South America merges with the fused North America-Europe and Asia-Pacific cluster. South Africa does not merge with the rest of the graph until a threshold beyond 120, at which point it allows for a representation of the global-scale infrastructure of Azure as a single connected component.

These observations are further refined by examining the curvature heatmap in the Azure column in panel (f) of Figure 4. The two dark red cells correspond to the edges between Tokyo-Quincy, and

[6] Animations of these manifold views can be found in our repository at https://github.com/Burdantes/Curvature-based-Analysis-of-Network-Connectivity-in-Private-Backbone-Infrastructures.

Mumbai-London, respectively. While the emergence of the Tokyo-Quincy link at threshold 90 is distinctly captured in panel (d), the appearance of the light-red colored cells next to the Tokyo-Quincy link indicates the slow emergence of connectivity between the North America-Europe and Asia-Pacific regions at higher thresholds. At the same time, we see that the Mumbai-London connection first emerges between threshold 90 (panel (d)) and threshold 120 (panel (e)). Also, the appearance of light-red colored cells surrounding the Mumbai-London link suggests the emergence of more connectivity for transporting traffic between Europe and Asia. Trying to place the Mumbai-London edge on a geographic map shows that any optimal cable route would have to cross Iran and the Black Sea. However, in view of existing geopolitical tensions afflicting those regions, such cable deployments are unlikely in the foreseeable future, resulting in the use of alternative routes that are longer and slower than their optimal counterparts. In this sense, just as we expect identified topographical features to cause sub-optimal cable deployment, our examination of Azure's global-scale infrastructure shows that the proposed curvature-based analysis is also capable of highlighting areas of geopolitical tension that can cause sub-optimal connectivity.

The threshold heatmap in panel (g) shows the emergence of three distinct well-performing regions: North America (upper left), Europe (center) and the Asia-Pacific region (lower right). At the same time, we observe that South America (Sao-Paulo) and Africa (Johannesburg) are only weakly connected to the rest of the network and as a result, experience overall sub-par performance. Moreover, while the Europe and North America clusters are composed of uniformly light-colored cells, the cells that make up the Asia-Pacific cluster are colored in different shades of green which in turn indicates that the different cities in this cluster connect via links that vary in performance. At the same time, the links between North America (or Europe) and the Asia-Pacific area emerge in a more homogeneous manner, indicating more consistent performance.

Finally, the Sankey diagram in panel (h) shows how the different regions that make up the Azure global network merge to form a single connected component as the performance threshold increases. This dynamic view of our curvature-based analysis of Azure's infrastructure shows that: *(1)* the Europe and North American clusters form early on at low thresholds, *(2)* the Asia-Pacific cluster to forms at higher thresholds, *(3)* the integration of Johannesburg and Sao-Paulo into a single component that constitutes the Azure global network takes place at yet higher thresholds, and *(4)* the emergence of the Tokyo-Quincy, London-Mumbai, Dublin-Boydton, and London-Ashburn links are critical connections in the Azure global network.

6.2 An Analysis of the Manifold View of Cloud Providers Global Connectivity

While Figure 4 illustrates the different steps of our overall methodology with examples of three large cloud provider networks, Figure 5 shows how the manifold views that we generate for each of these networks evolve as we increase the curvature thresholds from 30 to 120 in steps of size 30. In the following, we briefly describe the key features that are evident in these manifold views.

Starting with the curvature threshold 30 (row b), Figure 5 highlights the "islands" of positive curvature that emerge in areas of dense connectivity in North America, Europe and Asia/Australia. In each panel, we identify a strong positive curvature in Europe, curvature that is bifurcated between coasts in North America (except for Azure) and relatively more neutral curvature in Asia/Australia. For threshold 60 (row c), we see even higher peaks of positive curvature in the major regions and of negative curvature (*i.e.,* saddle-shaped surfaces) in the Azure, and AWS manifolds as connectivity between Europe and North America emerges. At threshold 90 (row d), we observe the merging of the European and North American islands along with the emergence of a negatively curved link between Europe and Asia/Australia for AWS. These features differ from the prior threshold and are similar to the features for threshold 120 (row e), where we now see links between Europe and Asia/Australia for the Azure network.

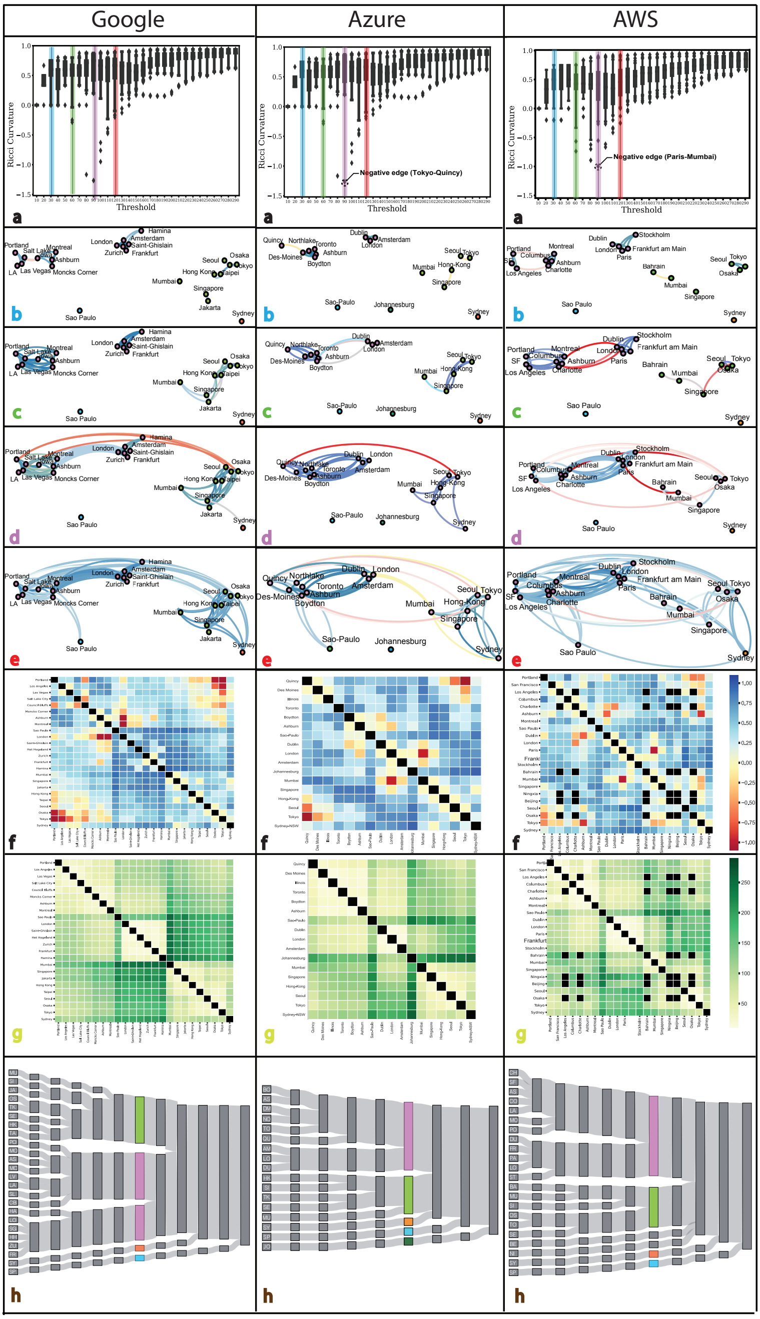


Fig. 4. Full pipeline for cloud provider infrastructures projected onto a world map at curvature thresholds (b) W=30, (c) X=60, (d) Y=90 and (e) Z=120. (f) and (g) are the Ricci curvature and threshold emergence heat maps. (h) are the Sankey diagrams.

The manifold views for the three cloud providers at a curvature threshold of 90 are shown in Figure 6. We focus on this threshold since it reveals significant features, commonalities and

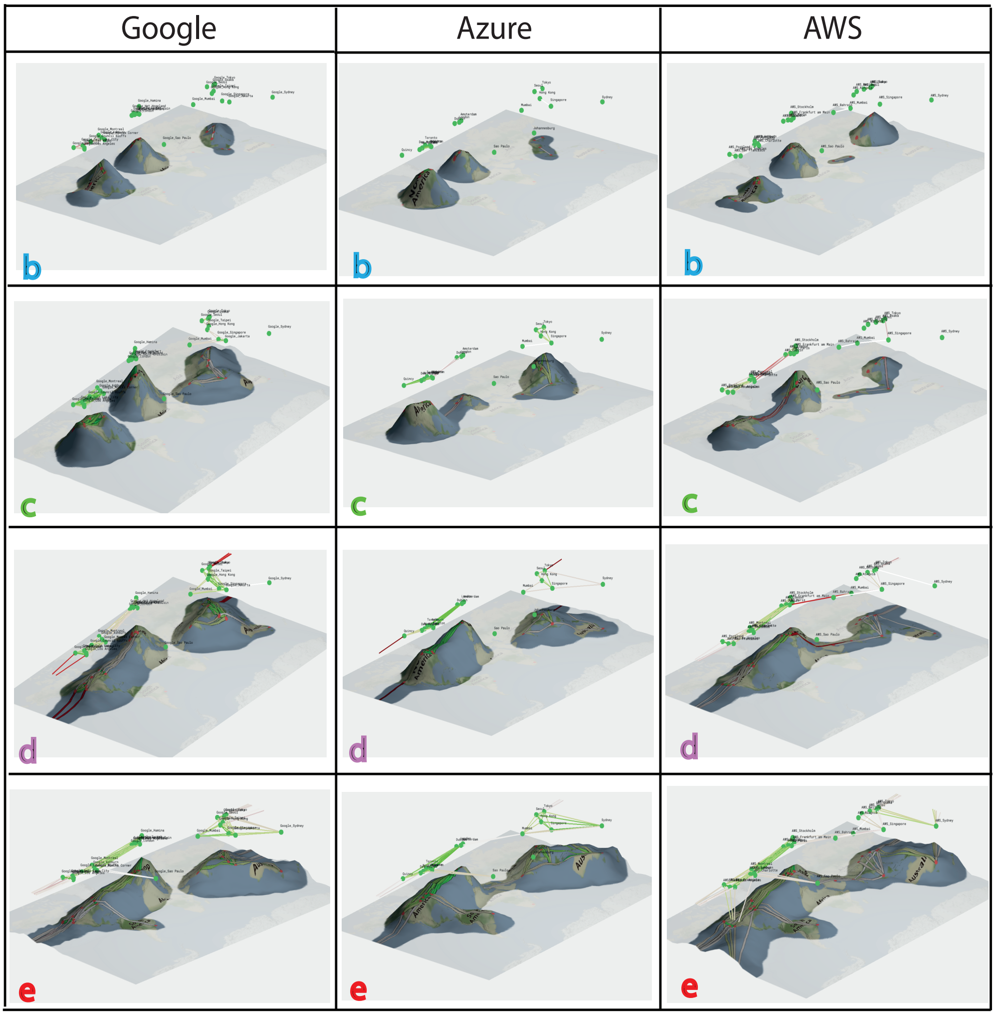


Fig. 5. Manifold views (with graphical connectivity shown above) for cloud provider infrastructures projected onto a world map at curvature thresholds (b) W=30, (c) X=60, (d) Y=90 and (e) Z=120.

differences in all three networks. Each sub-figure shows a 2-D graph of network connectivity above the surface for reference and a manifold projected onto a map of the earth below (where edges that stop at one side of the boundary of the surface wrap around the earth and continue on the other side of the surface's boundary).

The figure qualitatively confirms and precisely delineates three regions of rich connectivity at the global level: North America, Europe, and Asia/Australia, as shown by positively curved peaks above the plane. However, the boundaries of the richly connected regions, and the nature of the connectivity between them, varies considerably for the three networks. As indicated by the higher degree of positive curvature in Asia/Australia, the links between nodes in Google's network in that region are more efficiently connected. Apart from this difference, connectivity within richly connected regions for all networks is roughly comparable. Europe is the most positively curved region of the three (reflecting high connectivity over a comparatively compact region) while North

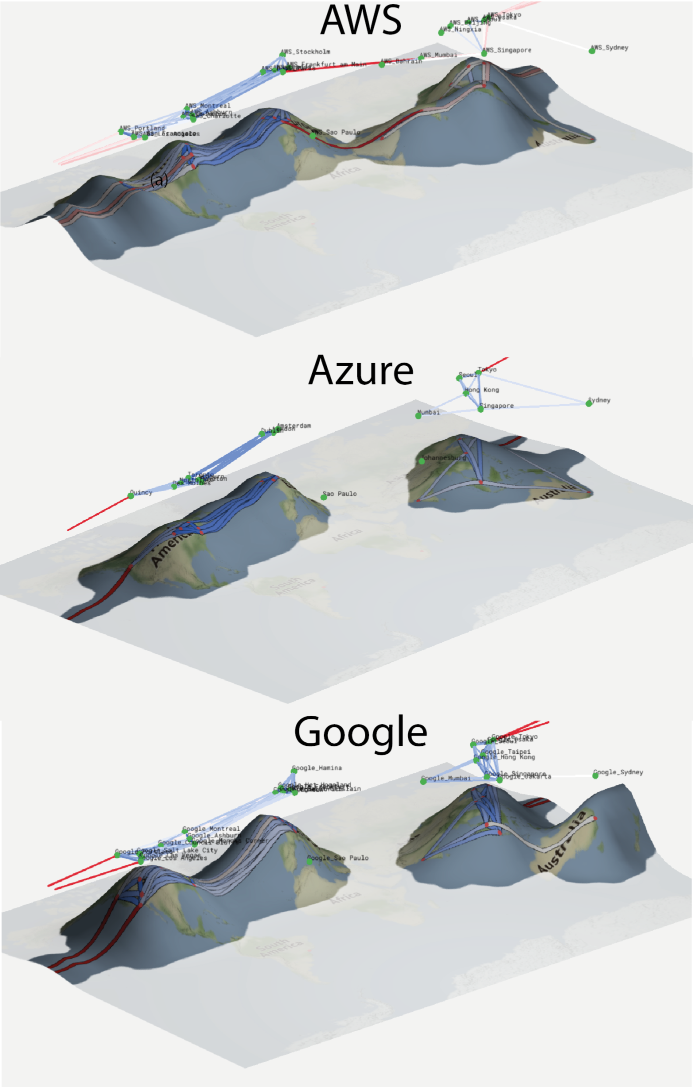


Fig. 6. Manifold view of AWS (top), Azure (middle), and Google (bottom) at threshold 90.

America shows positive curvature on the coasts, especially the East coast. In North America, AWS shows the most consistent positive curvature overall.

However, significant differences are apparent between cloud providers with respect to inter-region connectivity. Each positively curved region is connected to two others by approximately east-west pathways generating negative curvature. In general, trans-Atlantic connectivity between Europe and North America is fairly high-performing, with many edges at this performance level. At the other end of the spectrum, Europe-Asia connectivity is notably poor among all providers other than AWS. Furthermore, unlike the other providers, the AWS manifold includes transpacific paths with neutral curvature (as opposed to negative). The general lack of connectivity between Europe and Asia/Australia through the Mideast is surprising, since it is the shorter path and would appear to be feasible via multiple physical routes. However, geopolitical considerations have made high speed land-based cables difficult to site in this region [49], such that connectivity in this region is worse than, *e.g.,* across the Atlantic.

6.3 Manifolds Reflect the Delay Space

To illustrate how key network metrics are captured by our manifold view, we compute for selected pairs of vantage points (VPs) on a given manifold the geodesic distance as induced by the surface geometry of the manifold.[7] In this way, we identify both the distance between the cities hosting the VPs and a specific path between the VPs. We compute the geodesics on the manifold obtained at the threshold $\varepsilon = 90$ (as illustrated by the paths in Figure 6). We consider edges whose curvatures were identified to be particularly negative at that threshold and interesting cases to illustrate some of the key differences. We then construct linear predictors for the latency as a function of either *(i)* the GCD or *(ii)* the geodesic distance. We denote the prediction error for these two models as δ_{GCD} and δ_{Geo}. In both cases, we get rather good predictors with $R^2 > 0.9$ (only Google with GCD has a value lower close to 0.85) with a p-value smaller than 10^{-6} demonstrating that latency is indeed correlated with GCD and the geodesic distance.

Table 1 shows the superiority of the geodesic distance derived from the manifold as a predictor. The table entries provide the distance along the shortest-path in long-haul fiber infrastructure [29] for continental links and through submarines cables as provided in the Infrapedia database [2].[8]

For AWS, we observe that for all of the cities of interests, the estimator that uses geodesic distance performs better than the GCD and provides very accurate latency estimates, *i.e.,* less than 10 ms difference (in green) and decent latency estimates, *i.e.,* more than 10 ms difference (in yellow). Using the first two rows in the case of Google, we note that our technique also identifies links whose delays are influenced by long-haul fiber paths that are highly dilated compared to GCD. On the other hand, large delays tend to increase the latency estimation for well performing parts of the network (*e.g.,* Sydney-Las Vegas), but the geodesic distance-based estimators suffer less from this drawback. The last two rows illustrate that our predictor is less effective for mid-range links when considered at this threshold (in red). This could be resolved by either rescaling the positive bumps on the manifold or by repeating the geodesic computation between the cities on the manifold at the threshold when the edges have first emerged (*i.e.,* 70).

In summary, the geodesics are particularly useful for long-haul predictions. While our geodesics do not precisely exhibit the actual path taken by the data, the tendency to pass through saddle points highlights paths that lack direct physical connectivity. In general, geodesic distances are more appropriate estimators of network latency than GCD-based estimates, mainly because moving along the manifold rather than using line-of-sight results reflects actual delays more accurately. After all, the manifold view accounts for the deformation of the delay space that results from either the absence or abundance of infrastructure and manifests itself in the form of dilated or reduced delays.

7 RELATED WORK

Internet Latency Measurement. Internet latency based on RTT measurements has been the subject of many prior research studies including [11, 21, 29, 72, 97]. The utility of path latency measurements in research and in daily operations has led to the deployment of several specialized infrastructures (*e.g.*, see [5, 22, 42, 74] and also [14]). They are used to conduct on-going, ping-style

[7] To construct shortest path distances on the manifold, we employ an algorithm based on heat propagation – see [24] for details.

[8] A more detailed description of the technique is provided in Appendix C.2) (denoted d_{RD}) and along the great circle path (denoted d_{GCD}) at threshold 90 (when Asia-Pacific cluster arises).

Cloud	City A	City B	Latency	δ_{GCD}	δ_{Geo}	d_{RD}	d_{GCD}
Amazon	Sydney	Tokyo	104.6	-28.5	-1.2	10184	8147
	Ashburn	Sao Paulo	118.3	-12.2	3.0	9580	7664
	Dublin	Tokyo	202.0	39.0	-11.3	11980	9584
	Paris	Singapore	155.5	-27.0	-4.0	13415	10732
	Mumbai	Paris	104	-14.5	-6.0	8761	7009
Google	Osaka	London	221	55.4	5.7	12205	9499
	Mumbai	London	333	211.4	42.3	9247	7191
	Las Vegas	Sydney	142	-74.6	-48.7	16578	14320
	Seoul	Jakarta	106	14.5	40.2	6421	5293
	Portland	Frankfurt	124	-22.3	-42.4	9872	7834
Azure	Sydney	Tokyo	106.7	-26.3	-0.1	10184	8147
	Ashburn	Sao Paulo	117.8	-12.4	10.1	9580	7664
	Dublin	Tokyo	233.5	70.5	10.2	11980	9584
	Seoul	Amsterdam	220.5	75.7	25.6	11263	8516
	Singapore	Des-Moines	194.2	-59.5	-56.8	17872	14926

Table 1. Geodesic and GCD prediction of the latency for cities of interest and comparison between routing and geodesic distance. A negative value indicates that the predictor is underestimating the latency.

measurements between distributed sources and destinations and in most cases, they make their data openly available.

Network Embeddings. The challenges of scale and coverage in active probe-based latency measurement led to the development of methods for estimating distances between arbitrary nodes in the Internet using information from sets of landmarks (*e.g.,* [33, 36, 41, 63, 69, 82, 90]). These methods are referred to as *network embeddings*; that is, the mapping of node locations into a low-dimensional Euclidian space. Our work contrasts with these prior studies by highlighting the relationship between latency and the underlying physical infrastructure via curvature. While this relationship also includes aspects of the infrastructure's geographic location, in contrast to much prior work on IP geolocation (*e.g.,* see [30, 31, 37, 40, 83, 91]), our focus is on understanding connectivity between nodes.

Internet Topology. Accurate maps of the Internet's topological structure serve many purposes, and many prior studies have utilized `traceroute`-like tools or BGP data to map aspects such as *logical connectivity* between routers [16, 55], POPs [78, 81], Autonomous Systems [56, 96], or interconnections between networks [57]. Other studies have been concerned with creating accurate maps of *physical Internet infrastructure* (*e.g.*, see [3, 27]). While our goal of providing a useful perspective on Internet connectivity is similar to these studies, our curvature-based inference method is entirely different. At the same time, our work is informed by the recent study by Singla *et al.* [79] that identifies significant differences between physical distances between nodes and latency measurements.

Private Network Infrastructure. One of the early studies that characterized intra-cloud (private) and end host (public) network measurements for large cloud providers was conducted by Li *et al.* in [53]. Their results show highly stable TCP throughput rates between data centers for cloud providers but more variability in wide area latencies. A more recent report by ThousandEyes [51] provides details of a measurement study that characterizes network performance for Azure, Google and AWS from both intra-cloud and end user perspectives. Using measurements from VMs in data centers similar to ours, the key findings in that study include stable, high performance connectivity between data centers, variable performance from an end user perspective, and that the differences in global network layout can lead to degraded performance in certain instances. These and similar recent studies (*e.g.,* [12, 43, 95]) illustrate the importance of understanding the role of private networks in the Internet.

Manifolds and Curvature: The study of topological spaces including curvature and manifolds is an active area of mathematical research. Existing work in manifold learning aims to infer the manifold on which a set of data points lies by means of local neighborhood relations, attempting to preserve local distance while capturing global features. Prior work that has studied Internet delay space from this point of view includes [8], which focuses on inference of fractal dimension. The reported findings provide insights into the dimensionality of delay spaces, how they emerge from the data and how coordinate systems can be improved. In contrast, the authors in [68] apply manifold learning techniques to visualize changes in network traffic and as a means for anomaly detection. Moving beyond Euclidean spaces, the authors in [76] identify curvature as a characteristic of Internet paths and use this observation to generate a *hyperbolic embedding* that provides improved distance estimates. While each of the previous papers uses geometric views, none creates a complex manifold embedded in a higher dimensional space as we do this in this paper.

Prior attempts to infer complex manifolds from delay measurement alone have been highly abstract. The authors in [23] investigate the property of the latency space as a specific version of a distance realization problem and explore some latent statistical properties of the Internet. However they do not apply the approach to realistic measurements nor do they consider the implications of the resulting manifold shape. The authors in [10] also consider end-to-end measurements to infer structural features of the underlying topology but their effort is theoretical in nature. Other related works focus on the theoretical conditions required to be able to reconstruct such manifolds, for example [34, 62].

Ricci curvature has recently been applied to the analysis of networks. For example, the authors in [65] characterize several router- and AS-level data sets using Ricci curvature, and [87] leveraged Ricci curvature more broadly to detect the coarse geometry of networks. Our effort differs in the application of Ricci curvature to uncover specific topological features of the network using only delay and location.

8 SUMMARY AND OUTLOOK

This paper is motivated by recent developments that make it more onerous for third-party researchers to obtain data or perform experiments on connectivity of the increasingly important private infrastructures of the large cloud and content providers. The main contribution of this work is a newly proposed methodology for elucidating connectivity in known or inaccessible network infrastructures by leveraging light-weight and readily available latency measurements and applying advanced mathematical techniques from Riemannian geometry. Specifically, our approach is based on the application of Ricci curvature and its extension to discrete graphs known as Ollivier-Ricci curvature, which allows for a sophisticated encoding of a composite view of underlying physical infrastructure and deployed routing configurations into weighted graph representations of a set of measured network latencies. We apply our curvature-based network analysis the private connectivity fabric of large cloud providers, illustrate its ability to expose critical aspects of connectivity, and present examples of its practical relevance.

The graph models generated from a collection of simple pairwise latency measurements between nodes in a network (typically with unknown physical infrastructure) allow for a geometric representation as a complex topological space in the form of an idealized smooth manifold. Such manifolds offer unprecedented opportunities for characterizing a network's underlay that resulted in the measured latencies in the first place. Unraveling the presumed metric space structure of this complex geometric object and leveraging it to identify pertinent and otherwise hard-to-detect features of a given network's connectivity fabric and exploiting them to the fullest extent in practice looms as a promising avenue for future work.

In particular, given the availability of large corpora of open-sourced RTT measurement datasets (*e.g.* RIPE Atlas' latest longitudinal datasets of minRTT measurements [7]), a natural application of our methodology is to try to uncover similar manifold views for the provider-based best-effort ("public") Internet. However, extending our technique to adequately deal with the public Internet poses challenges. Indeed, since interdomain routing based on the Border Gateway Protocol (BGP) is latency agnostic (see [12] and references therein), any extension will necessarily have to account for the complexities caused by a BGP-based Internet routing fabric.

Acknowledgements We thank the anonymous reviewers and our shepherd Shaileshh Bojja Venkatakrishnan for their constructive feedback. We also thank Kave Salamatian for many conversations and helpful suggestions at the beginning of this project on using Ricci curvature for analyzing network measurements. We are grateful to Edith Ziang for helping with the proof in Appendix C. This material is based upon work supported by the National Science Foundation through awards CNS-1703592, CNS-2039146, and CNS-2106517. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work was also supported in part by a Google grant.

REFERENCES

[1] 2018. Reliance Communication plans undersea cable to meet data demands of Asia, Europe. https://tinyurl.com/ybjv7mdp. (2018).

[2] 2020. Infrapedia. https://www.infrapedia.com/. (2020).

[3] 2020. The Internet Topology Zoo. http://www.topology-zoo.org/. (2020).

[4] 2020. Network Service Tiers - Google Cloud. https://cloud.google.com/network-tiers/docs/overview. (2020).

[5] 2020. PingER: Ping End-to-end Reporting. http://www-iepm.slac.stanford.edu/pinger/. (2020).

[6] 2020. Telecom Ramblings: Metro Fiber Maps. https://www.telecomramblings.com/metro-fiber-maps/. (2020).

[7] Emile Aben. 2021. Latency Into Your Network - As Seen From RIPE Atlas. (2021). https://labs.ripe.net/author/emileaben/latency-into-your-network-as-seen-from-ripe-atlas/

[8] Bruno Abrahao and Robert Kleinberg. 2008. On the internet delay space dimensionality. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement.* 157–168.

[9] Najam Ahmad. 2016. Building one of the highest-capacity subsea cables in the Pacific. https://engineering.fb.com/connectivity/building-one-of-the-highest-capacity-subsea-cables-in-the-pacific/. (2016).

[10] Anima Anandkumar, Avinatan Hassidim, and J. Kelner. 2011. Topology discovery of sparse random graphs with few participants. *ArXiv* abs/1102.5063 (2011).

[11] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. 2001. Resilient Overlay Networks. In *Proceedings of ACM Symposium on Operating System Principles.*

[12] T. Arnold, E. Gurmericliler, G. Essig, A. Gupta, M. Calder, V. Giotsas, and E. Katz-Bassett. 2020. (How Much) Does a Private WAN Improve Cloud Performance?. In *Proceedings of IEEE INFOCOM.*

[13] T. Arnold, J. He, W. Jiang, M. Calder, Í. Cunha, V. Giotsas, and E. Katz-Bassett. 2020. Cloud Provider Connectivity in the Flat Internet. In *Proc. ACM IMC'20.*

[14] AT&T. 2020. Global IP Network Performance Measurements. https://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html. (2020).

[15] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. 2006. Avoiding Traceroute Anomalies with Paris Traceroute. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC '06).* Association for Computing Machinery, New York, NY, USA, 153–158. https://doi.org/10.1145/1177080.1177100

[16] Brice Augustin, Balachander Krishnamurthy, and Walter Willinger. 2009. IXPs: Mapped?. In *Proceedings of the ACM Internet Measurement Conference.*

[17] AWS. 2020. AWS Global Cloud Infrstructure. https://www.infrastructure.aws. (2020).

[18] R. Beverly. 2016. Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In *Proceedings of Internet Measurement Conference.*

[19] D. Bhattacherjee, S. Abdu Jyothi, I. Nadi Bozkurt, M. Tirmazi, W. Aqeel, A. Aguirre, B. Chandrasekaran, B. Godfrey, G. Laughlin, B. Maggs, and A. Singla. 2018. cISP: A Speed-of-Light Internet Service Provider. *arXiv* 1809.10897 (2018).

[20] Béla Bollobás, Svante Janson, and Oliver Riordan. 2010. The cut metric, random graphs, and branching processes. *Journal of statistical physics* 140, 2 (2010), 289–335.

[21] Ilker Bozkurt, Waqar Aqeel, Debopam Bhattacherjee, Balakrishnan Chandrasekaran, Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singlar. 2018. Dissecting Latency in the Internet's Fiber Infrastructure. *arXiv* 1811.10737 (2018).

[22] CAIDA. 2020. Archipelago Measurement Infrastructure. http://www.caida.org. (2020).

[23] Fan Chung, Mark Garrett, Ronald Graham, and David Shallcross. 2000. Distance Realization Problems with Applications to Internet Tomography. *J. Comput. System Sci.* 63 (12 2000), 432–448. https://doi.org/10.1006/jcss.2001.1785

[24] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. 2013. Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow. *ACM Transactions on Graphics* 32 (09 2013). https://doi.org/10.1145/2516971.2516977

[25] Alfredo Cuzzocrea, Alexis Papadimitriou, Dimitrios Katsaros, and Yannis Manolopoulos. 2012. Edge betweenness centrality: A novel algorithm for QoS-based topology control over wireless sensor networks. *Journal of Network and Computer Applications* 35, 4 (2012), 1210–1217.

[26] G. Detal, B. Hesmans, O. Bonaventure, Y. Vanaubel, and B. Donnet. 2013. Revealing Middlebox Interference with Tracebox. In *Proc. ACM IMC'13*.

[27] R. Durairajan, S. Ghosh, X. Tang, P. Barford, and B. Eriksson. 2013. Internet Atlas: A Geographic Database of the Internet. In *Proceedings of the ACM HotPlanet Workshop*.

[28] Ramakrishnan Durairajan, Joel Sommers, and Paul Barford. 2014. Layer 1-informed internet topology measurement. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. 381–394.

[29] R. Durairajan, J. Sommers, W. Willinger, and P. Barford. 2015. InterTubes: A Study of the US Long-haul Fiber-optic Infrastructure. In *Proceedings of ACM SIGCOMM*.

[30] E. Katz-Bassett and J. P. John and A. Krishnamurthy and D. Wetherall and T. Anderson and Y. Chawathe. 2006. Towards IP Geolocation Using Delay and Topology Measurements. In *Proceedings of the ACM Internet Measurement Conference*.

[31] B. Eriksson, P. Barford, B. Maggs, and R. Nowak. 2012. Posit: A Lightweight Approach for IP Geolocation. *ACM SIGMETRICS Performance Evaluation Review* (2012).

[32] ESRI. 2020. ArcGIS: The Mapping and Analytics Platform. https://www.esri.com/en-us/arcgis/about-arcgis/overview. (2020).

[33] F. Dabek and R. Cox and F. Kaashoek and R. Morris. 2004. Vivaldi: A Decentralized Network Coordinate System. In *Proceedings of ACM SIGCOMM*.

[34] Charles Fefferman, Sergei Ivanov, Matti Lassas, and Hariharan Narayanan. 2020. Reconstruction of a Riemannian manifold from noisy intrinsic distances. *SIAM Journal on Mathematics of Data Science* 2, 3 (2020), 770–808.

[35] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio. 2015. An Updated Performance Comparison of Virtual Machines and Linux Containers. In *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*.

[36] Paul Francis, Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang. 2001. IDMaps: A Global Internet Host Distance Estimation Service. *IEEE/ACM Transactions on Networking* 9, 5 (2001).

[37] P. Gill, Y. Ganjali, B. Wong, and D. Lie. 2010. Dude, Where's that IP?: Circumventing Measurement-based IP Geolocation. In *Proceedings of the USENIX Security Symposium*.

[38] Google. 2020. Curie subsea cable set to transmit to Chile, with a pit stop to Panama. https://cloud.google.com/blog/products/infrastructure/curie-subsea-cable-set-to-transmit-to-chile-with-a-pit-stop-to-panama. (2020).

[39] Google. 2020 (last updated). Virtual Private Cloud (VPC) Documentation: Traceroute to external IP addresses. https://cloud.google.com/vpc/docs/vpctraceroute. (2020 (last updated)).

[40] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. 2006. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Transactions on Networking* 14, 6 (2006).

[41] K. Gummadi, S. Saroiu, and S. Gribble. 2002. King: Estimating latency between arbitrary internet end hosts. In *Proceedings of the ACM Internet Measurement Workshop*.

[42] C. Guo, L. Yuan, D. Xiang, Y. Dang, R. Huang, D. Maltz, Z. Liu, V. Wang, B. Pang, H. Chen, Z.-W. Lin, and V. Kurien. 2015. Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis. In *Proceedings of the ACM SIGCOMM Conference*.

[43] O. Haq, M. Raja, and F. Dogar. 2017. Measuring and Improving the Reliability of Wide-Area Cloud Paths. In *Proceedings of WWW'17*.

[44] C.-Y. Hong, S. Kandula nd R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. 2013. Achieving High Utilization with Software-Driven WAN. In *Proceedings of the ACM SIGCOMM Conference*.

[45] Peter J Huber. 2004. *Robust statistics*. Vol. 523. John Wiley & Sons.

[46] V. Jacobson. 1989. traceroute. ftp://ftp.ee.lbl.gov/traceroute.tar.gz. (1989).

[47] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hoelzle, S. Stuart, and A. Vahdat. 2013. B4: Experience with a Globally-Deployed Software Defined WAN. In *Proceedings of the ACM SIGCOMM Conference*.

[48] M. Jimenez and H. Kwok. 2017. Building Express Backbone: Facebook's New Long-Haul Network . https://engineering.fb.com/data-center-engineering/building-express-backbone-facebook-s-new-long-haul-network/. (2017).

[49] R. Jones and D. Fitzgerald. 2020. Google Plans Fiber-Optic Network to Connect Via Saudi Arabia and Israel for First Time. https://www.wsj.com/articles/google-plans-fiber-optic-network-to-connect-via-saudi-arabia-and-israel-for-first-time-11606143590. (2020).

[50] C. Kaufmann. 2018. ICN - Akamai's Backbone. https://www.linx.net/wp-content/uploads/LINX101-Akamai-ICN-ChristianKaufmann.pdf. (2018).

[51] A. Kesavan. 2019. Comparing the Network Performance of AWS, Azure and GCP. https://pc.nanog.org/static/published/meetings/NANOG75/1909/20190218_Kesavan_Comparing_The_Network_v1.pdf. (2019).

[52] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. 2010. Hyperbolic geometry of complex networks. *Physical Review E* 82, 3 (2010), 036106.

[53] A. Li, X. Yang, S. Kandula, and M. Zhang. 2010. CloudCmp: Comparing Public Cloud Providers. In *Proceedings of the ACM Internet Measurement Conference*.

[54] L. Li, D. Alderson, W. Willinger, and J.C. Doyle. 2004. A First-principles Approach to Understanding the Internet's Router-level Topology. In *Proc. ACM SIGCOMM'04*.

[55] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. 2006. iPlane: An Information Plane for Distributed Services. In *Proceedings of the USENIX Symposium on Network Systems Design and Implementation*.

[56] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H Katz. 2003. Towards an Accurate AS-Level Traceroute Tool. In *Proceedings of ACM SIGCOMM*.

[57] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, Claffy KC, and Smith M. Jonathan. 2018. Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale. In *Proceedings of the ACM Internet Measurement Conference*.

[58] Microsoft. 2015. Microsoft Invests in Subsea Cables to Connect Datacenters Globally. https://www.esri.com/en-us/arcgis/about-arcgis/overview. (2015).

[59] Microsoft. 2020. Azure Microsoft Global Network Map. https://docs.microsoft.com/en-us/azure/networking/microsoft-global-network. (2020).

[60] Microsoft. 2021. Azure Virtual Network frequently asked questions (FAQ). https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-faq. (2021).

[61] R. Motamedi, R. Rejaie, and W. Willinger. 2014. A survey of techniques for internet topology discovery. In *IEEE Communications Survey & Tutorials*.

[62] Hariharan Narayanan and Sanjoy Mitter. 2010. Sample complexity of testing the manifold hypothesis. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 2*. 1786–1794.

[63] E. Ng and H. Zhang. 2002. Predicting Internet Network Distance with Coordinates-based Approaches. In *Proceedings of IEEE INFOCOM*.

[64] C. Ni, Y. Lin, F. Luo, and J. Gao. 2019. Community Detection on Networks with Ricci Flow. *Scientific Reports* 9, 9984 (2019).

[65] Chien-Chun Ni, Yu-Yao Lin, Jie Gao, Xianfeng David Gu, and Emil Saucan. 2015. Ricci Curvature of the Internet Topology. *arXiv* 1501.04138 (2015).

[66] Yann Ollivier. 2009. Ricci curvature of Markov chains on metric spaces. *Journal of Functional Analysis* 256, 3 (2009), 810–864.

[67] Y. Ollivier. 2011. A Visual Introduction to Riemannian Curvatures and Some Discrete Generalizations. Analysis and Geometry of Metric Measure Spaces: Lecture Notes of the 50th Seminaire de Mathematiques Superieures (SMS). (2011).

[68] N. Patwari, A. Hero, and A. Pacholski. 2008. Manifold learning visualization of network traffic data. In *Proceedings of the ACM Internet Measurement Conference*.

[69] Marcelo Pias, Jon Crowcroft, Steve Wilbur, Tim Harris, and Saleem Bhatti. 2003. Lighthouses for Scalable Distributed Location. In *Proceedings of the USENIX International Workshop on Peer to Peer Systems*.

[70] John W Pinney and David R Westhead. 2006. Betweenness-based decomposition methods for social and biological networks. *Interdisciplinary statistics and bioinformatics* 25 (2006), 87–90.

[71] K. Putzier. 2020. Property Investors See Fiber-Optic Cables as 'Railroads of the Future'. (2020).

[72] R. Mahajan and M. Zhang and L. Poole and V. Pai. 2008. Uncovering Performance Differences in Backbone ISPs with Netdiff. In *Proceedings of the USENIX Symposium on Network Systems Design and Implementation*.

[73] Corinne Reichert. 2018. Hawaiki subsea cable goes live. https://www.zdnet.com/article/hawaiki-subsea-cable-goes-live/. (2018).

[74] RIPE. 2020. RIPE Atlas. https://atlas.ripe.net. (2020).

[75] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng. 2017. Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In *Proceedings of the ACM SIGCOMM Conference.*

[76] Y. Shavitt and T. Tankel. 2004. On the Curvature of the Internet and its Usage for Overlay Construction and Distance Estimation. In *Proceedings of IEEE INFOCOM.*

[77] R. Shea, F. Wang, H. Wang, and J. Liu. 2014. A Deep Investigation Into Network Performance in Virtual Machine Based Cloud Environments. In *Proceedings of IEEE INFOCOM.*

[78] Rob Sherwood, Adam Bender, and Neil Spring. 2008. Discarte: A Disjunctive Internet Cartographer. In *Proceedings of ACM SIGCOMM.*

[79] Ankit Singla, Balakrishnan Chandrasekaran, P Brighten Godfrey, and Bruce Maggs. 2014. The Internet at the Speed of Light. In *Proceedings of the ACM ACM Workshop on Hot Topics in Networks.*

[80] J. Sommers, P. Barford, and B. Eriksson. 2011. On the prevalence and characteristics of MPLS deployments in the open Internet. In *Proc. of the ACM Internet Measurement Conference (IMC'11).*

[81] Neil Spring, Ratul Mahajan, and David Wetherall. 2002. Measuring ISP Topologies with Rocketfuel. In *Proceedings of ACM SIGCOMM.*

[82] Liying Tang and Mark Crovella. 2003. Virtual landmarks for the Internet. In *Proceedings of the ACM Internet Measurement Conference.*

[83] V. Padmanabhan and L. Subramanian. 2001. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceedings of ACM SIGCOMM.*

[84] A. Vahdat, D. Clark, and J. Rexford. 2015. A Purpose-built Global Network: Google's Move to SDN. *ACMQueue* 13, 8 (2015).

[85] A. Vogel, D. Griebler, C. Schepke, and L. Fernandes. 2017. An Intra-Cloud Networking Performance Evaluation on CloudStack Environment. In *IEEE 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing.*

[86] G. Wang and T. Ng. 2010. The Impact of Virtualization on Network Performance of Amazon EC2 Data Center. In *Proceedings of IEEE INFOCOM.*

[87] M. Weber, J. Jost, and E. Saucan. 2018. Detecting the coarse geometry of networks. In *Proceedings of NeurIPS 2018 Workshop.*

[88] W. Willinger, D. Alderson, and J. C. Doyle. 2009. Mathematics and the Internet: A Source of Enormous Confusion and Great Potential. In *Notices of the AMS.*

[89] F. Wohlfart, N. Chatzis, C. Dabanoglu, G. Carle, and W. Willinger. 2018. Leveraging Interconnections for Performance: The Serving Infrastructure of a Large CDN. In *Proceedings of the ACM SIGCOMM Conference.*

[90] B. Wong, A. Slivkins, and E. Sirer. 2005. Meridian: A Lightweight Network Location Service Without Virtual Coordinates. In *Proceedings of ACM SIGCOMM.*

[91] B. Wong, I. Stoyanov, and E. Sirer. 2007. Octant: A Comprehensive Framework for the Geolocation of Internet Hosts. In *Proceedings of the USENIX Symposium on Network Systems Design and Implementation.*

[92] M. Xavier, M. Neves, F. Rossi, T. Ferreto, T. Lange, and C. De Rose. 2013. Performance Evaluation of Container-based Virtualization for High Performance Computing Environments. In *IEEE 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing.*

[93] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, V. Lin, C. Rice, B. Rogan, A. Singh, B. Tanaka, M. Verma, M. Tariq, M. Tierney, D. Trumic, V. Valancius, P. Sood, C. Ying, M. Kallahalla, B. Koley, and A. Vahdat. 2017. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *Proceedings of the ACM SIGCOMM Conference.*

[94] B. Yeganeh, R. Durairajan, R. Rejaie, and W. Willinger. 2019. How Cloud Traffic Goes Hiding: A Study of Amazon's Peering Fabric. In *Proceedings of the Internet Measurement Conference (IMC).*

[95] B. Yeganeh, R. Durairajan, R. Rejaie, and W. Willinger. 2020. A First Comparative Characterization of Multi-cloud Connectivity in Today's Internet. In *Proceedings of PAM'20.*

[96] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. 2005. Collecting the Internet AS-Level Topology. *ACM SIGCOMM Computer Communications Review* 35, 1 (2005).

[97] Yin Zhang and Nick Duffield. 2001. On the Constancy of Internet Path Properties. In *Proceedings of the ACM Internet Measurement Workshop.*

APPENDIX

A REPRODUCIBILITY

All datasets and source code described in this paper can be found in https://github.com/Burdantes/Curvature-based-Analysis-of-Network-Connectivity-in-Private-Backbone-Infrastructures.

B DATA CENTER LOCATIONS

Table 2 lists the locations of the data centers that were considered in our study of private Internet infrastructure. The table indicates the city that is associated with data center deployment, but does not indicate where there are multiple data centers in a single location.

C ALTERNATIVE GRAPH AND DISTANCE METRICS

C.1 Ricci Curvature vs. Standard Graph Metrics:

Edge betweenness and the cut metric are two standard metrics used to characterize edge importance in a graph (for details, see [25, 70] for betweenness and [20] for the cut metric). A key strength of our technique lies in its ability to focus on a given edge and track the evolution of the curvature of that edge as the threshold increases, resulting in changes to the graph. In this section, we argue that the cut metric and edge betweenness are less effective for this task, because both metrics can be highly unstable and sensitive to the addition or deletion of specific edges. Edge curvature is a more robust alternative, as it focuses solely on local changes. Moreover, Ricci curvature is able to uncover the same features that edge betweenness and the cut metric can identify.

Cut Metric: For a connected graph $G = (V, E)$, a cut S between two disjoint subsets of vertices A and B is a collection of edges s such that in $G_{\text{cut}} := (V, E - S)$, there is no path $a \mapsto b$ for any two nodes $a \in A$ and $b \in B$. A cut is said to be minimal when the cardinality of S is minimal. This cardinality is then called the minimal cut number c. The minimal cut metric is relative to the choice of two disjoint sets A and B. When selecting A and B as two different continents, for example, Ricci curvature and the minimal cut number are closely related. Specifically, negative Ricci curvature induces a small minimal cut number since there is very little redundancy in connecting the neighborhoods. Selecting A and B in the same continent, for example, we note an inverse trend with positive Ricci curvature corresponding to a large minimal cut number as there exist many redundancies to link the different sets A and B. Furthermore, the cut metric focuses on connected components and the existence of any path, which in turn allows for convoluted paths to be considered.

Betweenness centrality: The betweenness centrality of an edge e, written as $\beta(e)$, is defined by the expression:

$$\beta(e) = \sum_{s,t \in V, s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(e)$ is the number of those paths that cross e. While both Ricci curvature and betweenness centrality are metrics defined on shortest paths, the betweenness metric of a single edge is a function of all the shortest paths in the graph; in contrast, Ricci curvature only considers paths in the neighborhood of the edge.

First, the important thresholds identified by betweenness centrality correspond (*i.e.* identifying highly positive betweenesss centrality edges relate back to negative Ricci curvature) to those established by the Ricci curvature. This means that had our analysis been based on using betweenness centrality, it would have yielded similar conclusions to our findings based on edge curvature. However, it is important to point out that the impact of the addition of an edge to the betweenness

Table 2. Locations of hosts used to collect minRTT data for our study.

Abbrev	Location	AWS	Azure	GCP	Public
AM	Amsterdam, Netherlands		x		x
AS	Ashburn, Virginia	x	x	x	
BA	Bahrain	x			x
BE	Beijing, China	x			
BO	Boydton, Virginia		x		
BR	Brussels, Belgium				x
CA	California				x
CB	Council Bluffs, Iowa			x	
CH	Charlotte, NC	x			
CO	Columbus, Ohio	x			
DM	Des Moines, Iowa		x		
DO	Doha, Qatar				x
DU	Dublin, Ireland	x	x		x
FR	Frankfurt, Germany	x		x	x
HA	Hamina, Finland			x	
HH	Het Hogeland, Netherlands			x	
HK	Hong Kong, China		x	x	x
IA	Iowa				x
JA	Jakarta, Indonesia			x	x
JO	Johannesburg, South Africa		x		x
KL	Kuala Lumpur, Malaysia				x
KY	Kymenlaakso, Finland				x
LA	Los Angeles, California	x		x	
LO	London, Great Britain	x	x	x	x
LV	Las Vegas, Nevada			x	
MC	Moncks Corner, SC			x	
MO	Montreal, Canada	x		x	
MU	Mumbai, India	x	x	x	x
NC	North Carolina				x
NH	North Holland				x
NI	Ningxia, China	x			x
NO	Northlake, Illinois		x		
OH	Ohio				x
OR	Oregon				x
OS	Osaka, Japan	x		x	x
PA	Paris, France	x			x
PO	Portland, Oregon	x		x	
QU	Quebec, Canada				x
QY	Quincy, Washington (state)		x		
SC	South Carolina				x
SE	Seoul, South Korea	x	x	x	x
SF	San Francisco, California	x			x
SG	Saint-Ghislain, Belgium			x	
SI	Singapore, Singapore	x	x	x	x
SL	Salt Lake City, Utah			x	
SM	Stockholm, Sweden	x			x
SP	São Paulo, Brazil	x	x	x	x
SY	Sydney, Australia	x	x	x	x
TA	Taipei, Taiwan			x	x
TK	Tokyo, Japan	x	x	x	x
TO	Toronto, Canada		x		x
UT	Utah				x
VA	Virginia				x
WA	Washington (state)				x
ZU	Zürich, Switzerland			x	x

score of another edge cannot be upper bounded by a constant bound. Adding a single edge can shift a large portion of the shortest paths from crossing one edge to another one, suggesting that betweenness centrality might be subject to a large increase or decrease of its value. In light of this lack of stability and of the noisy character of our measurements (*e.g.,* erroneous latency measurements, wrong geolocations...) the betweenness centrality metric can be more sensitive to noise and outliers. On the other hand, the Ricci curvature of a given edge is necessarily increasing as the threshold increases. This increase is bounded by a function of the number of edges that are added in a given neighborhood. Below, we provide a proof of our assertion about the stability of the Ricci curvature, more specifically, we show that the total change in Ricci curvature of adding any edge is at most **6**. This upper bound is not sharp and we believe that there is room for improvement.

Before doing so, we start from the standard notion of adversarial perturbation in robust statistics (see [45] for further details on this notion) to quantify the maximum effect on the centrality measure of a single edge addition or removal. The breakdown point of an estimator is the proportion of arbitrarily large observations the estimator can sustain before giving an arbitrarily large result. In the context of this paper, we use the same idea: the estimator is the Ricci curvature κ of an edge $\{x, y\}$, where “observations" correspond to the edges adjacent to x or y. An adversarial new edge has bounded, local effect on the Ollivier-Ricci curvature in an unweighted graph. In particular, a new edge between nodes x and z can only affect the Ricci curvature of edges connected to neighbors of x and z. We study a fixed edge between nodes x and y and consider the three different cases of nearby edge additions that could affect $\kappa(x, y)$. We denote κ' as the Ricci curvature after the edge addition.

A first localized bound: Consider a graph with two nodes x, y connected with an edge between them. In the general scenario, x (respectively y) has n (respectively m) neighbors that are strongly unshared with y (x), and there are ℓ weakly unshared or overlapping neighbors between x, y. (See figure 7 for illustration). We compute the optimal transport between x, y:

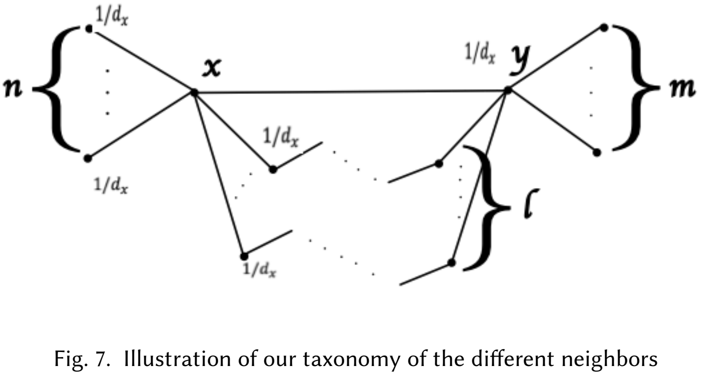


Fig. 7. Illustration of our taxonomy of the different neighbors

The optimal transport between x, y with the distribution as defined in 3, where $dx = n + \ell + 1$ and $dy = m + \ell + 1$, is

$$W(x,y) = \frac{n}{d_x} + \frac{m}{d_y} + \sum_{i=1}^{\ell} \min\left(\frac{1}{d_y}, \frac{1}{d_x}\right) \min(k_i, 3) + \sum_{i=1}^{\ell} \left|\frac{1}{d_y} - \frac{1}{d_x}\right| \min(k_i + 1, 2)$$
$$+ \frac{n+1}{d_x} - \frac{1}{d_y} - \mathbf{1}_{\{n>m\}} \cdot \ell \left(\frac{1}{d_y} - \frac{1}{d_x}\right).$$

Let x, z be the new edge added. We consider three cases: either z is strongly unshared with y (case 1), z is weakly shared with y (case 2) or z is added between neighbors of x, y (directly shared).

Case 1: this is equivalent to letting $n \mapsto n+1$. By extension, we also have $d_x \mapsto d_x + 1$ in the above equation. We get $W' - W$ to be

$$(W' - W)(x,y) = \frac{n+1}{d_x+1} - \frac{n}{d_x} \tag{2}$$
$$+ \mathbf{1}_{m \le n} \left(\frac{1}{d_x} - \frac{1}{d_x+1}\right) \cdot \sum_{i=1}^{\ell} \min(k_i, 3) \tag{3}$$
$$+ \sum_{i=1}^{\ell} \left[\left(\left|\frac{1}{d_y} - \frac{1}{d_x+1}\right| - \left|\frac{1}{d_y} - \frac{1}{d_x}\right|\right) \min(k_i + 1, 2)\right] \tag{4}$$
$$+ \frac{n+2}{d_x+1} - \frac{n+1}{d_x} \tag{5}$$
$$- \mathbf{1}_{\{n+1>m\}} \cdot \ell \left(\frac{1}{d_y} - \frac{1}{d_x+1}\right) - \mathbf{1}_{\{n>m\}} \cdot \ell \left(\frac{1}{d_y} - \frac{1}{d_x}\right). \tag{6}$$

We consider each of these terms separately and determine bounds individually over each term. We will consider the case where $\ell \ge 1$ since some terms depend on ℓ and become trivial when $\ell = 0$. Since all the terms are non-increasing with respect to d_x, we can simply use $\ell = 1$ and $d_x = 3$ to get the best nontrivial bounds. The bounds for $\ell \ge 1$ are at least as good as the ones for $\ell = 0$.

In particular, we have that when $\ell \ge 1$, $d_x \ge 2$,

$$(2) = \frac{\ell + 2}{d_x(d_x+1)} \in (0, 1/2]$$

$$(5) = \frac{\ell + 1}{d_x(d_x+1)} \in (0, 1/3]$$

where the upper bounds are achieved when $d_x = 1$, the minimum allowable number of neighbors.

Also note that

$$(*) = \left|\frac{1}{d_y} - \frac{1}{d_x+1}\right| - \left|\frac{1}{d_y} - \frac{1}{d_x}\right| = \begin{cases} \frac{1}{d_x} - \frac{1}{d_x+1} & \text{if } n \ge m \\ \frac{1}{d_x+1} - \frac{1}{d_x} & \text{if } n < m \end{cases}$$

So

$$(4) = \sum_{i=1}^{\ell} (*) \min(k_j + 1, 2) \quad \le 2\ell(*) \quad \in \begin{cases} (0, 1/6] & \text{if } n \ge m \\ [-1/6, 0) & \text{if } n < m. \end{cases}$$

Furthermore,

$$(3) \leq \begin{cases} \left(\frac{1}{d_x} - \frac{1}{d_x+1}\right) 3\ell & \text{if } n \geq m \\ 0 & \text{if } n < m \end{cases} \in \begin{cases} (0, 1/2] & \text{if } n \geq m \\ \{0\} & \text{if } n < m \end{cases}$$

Finally,

$$(6) = \begin{cases} -\ell\left(\frac{1}{d_x} - \frac{1}{d_x+1}\right) & \text{if } n = m \\ -\ell\left(\frac{1}{d_x+1} - \frac{1}{d_x}\right) & \text{if } n > m \\ 0 & \text{if } n+1 < m \end{cases} \in \begin{cases} [-1/6, 0) & \text{if } n = m \\ (0, 1/6] & \text{if } n > m \\ \{0\} & \text{if } n < m \end{cases}$$

Combining all these terms, we obtain

$$W' - W \in \begin{cases} (0, \ \ 5/3] & \text{if } n > m \\ (-1/6, \ \ 3/2) & \text{if } n = m \\ (-1/6, \ \ 5/6] & \text{if } n < m. \end{cases} \tag{7}$$

Bounds on $\kappa' - \kappa$ follow easily from here, since $d(x, y) = 1$ regardless.

$$(\kappa' - \kappa)(x, y) = -(W' - W)(x, y).$$

Case 2: z is weakly unshared or shared with y. The optimal adversarial placement (decrease W the most) is to add an edge (x, c_j) where $j = \text{argmax}_i(K_i)$. Then

$$\begin{aligned} (W' - W)(x, y) &= \min(1/d_x, 1/d_y)(\min(k_j, 2) - \min(k_j, 3)) \\ &\quad + \mathbf{1}_{n>m}\left|\frac{1}{d_y} - \frac{1}{d_x}\right|\left(\min(k_j + 1, 1) - \min(k_j + 1, 2)\right). \\ &\leq -\min(1/d_x, 1/d_y) - \left|\frac{1}{d_y} - \frac{1}{d_x}\right| \\ &= -\frac{1}{d_x} - \left(\frac{1}{d_y} - \frac{1}{d_x}\right) \\ &= -\frac{1}{d_y} \end{aligned}$$

where the upper bound is reached only if $k_j \geq 3$, and $n > m$, which gives the second equality. Furthermore, $d_y \geq 2$ since $\ell \geq 1$ (since $k_j \geq 1$ for some j). So

$$(W' - W)(x, y) \in \left[-\frac{1}{2}, 0\right].$$

Case 3: an edge is added between neighbors of x, y. The worst case is when the neighbor of x is strongly unshared with y, and vice versa. The difference here is

$$(W' - W)(x, y) = \begin{cases} -2 + 2/dx + 2/dy & \text{if } \frac{n-1}{n+1} > \frac{1}{m+1} \\ -2\min(1/dy, 1/dx) & \text{if } \frac{n-1}{n+1} < \frac{1}{m+1} \\ 2 - 3/dx & \text{if } \frac{n-1}{n+1} = \frac{1}{m+1} \end{cases}$$

Bounds on global distribution

Cases 1 and 2 above cover the possible change $(W' - W)(x, y)$ and thus $(\kappa' - \kappa)(x, y)$ after a new edge is introduced from x to any other node. From this, we can bound the effect of perturbations of a graph (addition/removal of edges) on the neighboring edges. Case 3 covers the possible change in $(\kappa' - \kappa)(x, y)$ when a new edge is added to neighbors of x.

Also, if a new edge were introduced between non-neighboring nodes along the path $b_i, \ldots, c_i$, there would be no effect on $W(x, y)$ since $d(b_i, c_i) \leq 3$. In general, if a new edge is added to a node not neighboring either x or y, there will be no effect on $Ri(x, y)$.

With this information, we can bound the effect on the graph's overall distribution of κ resulting from a single edge addition. First, only the edges adjacent to the new edge can be affected, with bounds given as above. In particular, the total effect of each term is bounded:

$$d_x \cdot ((2) + (5)) = \frac{2\ell + 3}{d_x + 1} \in (0, 2] \tag{8}$$

For $d_x \geq 2$,

$$d_x \left(\frac{1}{dx} - \frac{1}{dx + 1} \right) \in (0, 1/3]$$

leading to get

$$d_x(4) \in \begin{cases} (0, 1/3] & \text{if } n \geq m \\ [-1/3, 0) & \text{if } n < m \end{cases} \tag{9}$$

and

$$d_x(3) \leq \begin{cases} \frac{3\ell}{d_x+1} & \text{if } n \geq m \\ 0 & \text{if } n < m \end{cases} \in \begin{cases} (0, 3) & \text{if } n \geq m \\ \{0\} & \text{if } n < m \end{cases} \tag{10}$$

So the total change in Ricci curvature, graph-wide, is bounded by

$$\left| \sum_{e \in E} (\kappa' - \kappa)(e) \right| \leq 5 + 2/3 \tag{11}$$

where the sum over edges $e \in E$ is only over the neighboring edges to the new node. By extension, the overall distribution is unlikely to change, specifically if there is a large number of edges.

C.2 Routing distance

We define the *physical path distance or routing distance* between nodes as the sum of the lengths of individual fibers that make up the shortest (symmetric) physical path connecting the source and destination RIPE Atlas anchors and VMs in datacenters used in our latency measurements. Although we cannot know the exact physical paths traversed by our latency probes, routing decisions cannot define a path *shorter* than what is feasible according to the existing physical Internet infrastructure identified in our data sources. To obtain consistent estimates of end-to-end physical path distances, we utilize four different databases of physical infrastructure deployments [2, 3, 6, 27] and consider three different cases.

First, noting that a ping RTT of 0.5 ms corresponds to nodes within about 50km of each other, we consider all nodes within a 50km radius (*i.e.,* within the same metropolitan area) as having insignificant latency differences and therefore the same physical path distance to nodes outside of this area. Second, to measure the path distance between nodes in different metropolitan areas on the same land mass (*e.g.,* all nodes in North America), we identify all possible fiber paths from our infrastructure databases that connect metro areas, use a geographic information system

(ArcGIS [32]) to measure the distance of each possible fiber path between the centers of the metropolitan areas, and then choose the path with the shortest distance. Third, to measure the routing distance between nodes separated by an ocean, we identify the undersea cables connecting those continents, reference their lengths in the Infrapedia database [2], and select the cable with the shortest length. We then determine the nearest node to each landing location, measure the distance from the node to the landing location on both sides of the cable using the shortest path methodology described above, and then sum these three distances to estimate the path distance between the two nodes.

Experimenting with routing-based vs GCD-based distance metrics (not shown here) allows us to conclude that to obtain an accurate coarse-grained view of the structure of the physical underpinning of a provider network, the use of the GCD metric in conjunction with the Ricci curvature as mathematical tool is highly effective and conclusive. The use of more realistic metrics such as routing distance comes in handy to further confirm already inferred structure or zoom in on a given region to infer more fine-grained structure but requires a much higher overhead.

Received October 2021; revised December 2021; accepted January 2022