It Takes Two to Tango: Cooperative Edge-to-Edge Routing

Henry Birge-Lee
Princeton University
birgelee@princeton.edu

Maria Apostolaki
Princeton University
apostolaki@princeton.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

ABSTRACT

In their unrelenting quest for lower latency, cloud providers are deploying servers closer to their customers and enterprises are adopting paid Network-as-a-Service (NaaS) offerings with performance guarantees. Unfortunately, these trends contribute to greater industry consolidation, benefiting larger companies and well-served regions while leaving little room for smaller cloud providers and enterprises to flourish. Instead, we argue that the public Internet could offer good enough performance, if only edge networks could work together to achieve better visibility and control over wide-area routing. We present Tango, a cooperative architecture where pairs of edge networks (e.g., access, enterprise, and data-center networks) collaborate to expose more wide-area paths, collect more accurate measurements, and split traffic more intelligently over the paths. Tango leverages programmable switches at the borders of the edge networks, coupled with techniques to coax BGP into exposing more paths, without requiring support from end hosts or intermediate ASes. Experiments with our preliminary Tango deployment (using IPv6 addresses and the Vultr cloud provider) show that Tango could offer much greater visibility and control over wide-area routing, allowing the public Internet to meet the needs of many modern networked applications.

CCS CONCEPTS

• **Networks** → *Network layer protocols*; **Programmable networks**; *Naming and addressing*; Cloud computing; **Network monitoring**;

KEYWORDS

Network Measurement, Multipath Routing, BGP, SDN

ACM Reference Format:
Henry Birge-Lee, Maria Apostolaki, and Jennifer Rexford. 2022. It Takes Two to Tango: Cooperative Edge-to-Edge Routing. In *The 21st ACM Workshop on Hot Topics in Networks (HotNets '22), November 14–15, 2022, Austin, TX, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3563766.3564107

HotNets '22, November 14–15, 2022, Austin, TX, USA

ACM ISBN 978-1-4503-9899-2/22/11.
https://doi.org/10.1145/3563766.3564107

1 INTRODUCTION

Modern networked services—such as interactive applications (e.g., online gaming and video conferencing) and cyberphysical systems (e.g., self-driving cars and factory automation)—increasingly rely on predictable, low-latency communication. Yet, today's Internet is not up to the task. In response, large cloud providers are adopting edge computing, where cloud servers are deployed in, or near, access networks to be closer to their users. However, edge computing incurs significant costs to deploy, manage, and upgrade servers at many locations. Alternatively, enterprises can adopt Network-as-a-Service (NaaS) models such as "connectivity cloud" for reliable, high-performance communication with their cloud-hosted resources, but at the cost of installing specialized equipment and using dedicated network bandwidth.

These solutions work by *side-stepping* the long-standing performance and reliability problems with the Internet core. Instead, we argue that the public Internet can be made to support the low latency and high reliability that modern networked services need, at much lower cost. Working with the public Internet would enable a much broader collection of cloud providers and enterprises to flourish, including smaller organizations and underserved regions. Yet, despite years of effort, wide-area traffic delivery still faces significant limitations. We believe the key is to enable pairs of edge networks (e.g., an access network and a cloud datacenter) to *cooperate* to control how their traffic traverses the public Internet. Working together, a pair of edge networks can:

Expose greater path diversity: The Border Gateway Protocol (BGP) typically selects a single best path for each IP prefix. Edge networks with few neighbors learn few paths, and these paths may not offer the best performance. In contrast, cooperating edge networks can expose additional paths to each other.

Collect more accurate measurements: Traditional end-to-end performance measurements are notoriously noisy (due to variable loss and intra-domain delay), and round-trip measurements make it hard to understand one-way path performance. In contrast, cooperating edge networks can collect one-way measurements of the wide-area paths between them.

Make better routing decisions: Armed with greater path diversity and more accurate path-level measurements, edge networks can make more informed routing decisions. Plus, by associating a tunnel with each wide-area path, the cooperating edge networks can exert more direct control over which of these paths carries the traffic from one edge network to another.

Plus, by working together, the edge networks can achieve these goals without support from the end hosts or intermediate Autonomous Systems (ASes), greatly simplifying incremental deployment and reducing cost.

In this paper, we present Tango, an architecture that enables edge networks to cooperate in wide-area traffic delivery. For example, a cloud provider could offer Tango as a service to customer access or enterprise networks, or a distributed enterprise could run Tango between its multiple locations. Tango runs at the border switches of each of the edge networks, and leverages recent advances in programmable data planes to collect one-way performance measurements and tunnel traffic over specific wide-area paths to the other edge networks. The Tango edge networks also adopt techniques at the edge to expose greater path diversity, including using multiple IPv6 subnets and BGP communities to reveal multiple underlying wide-area paths to each other. We describe our experiences deploying and evaluating an initial Tango prototype using the Vultr [3] cloud provider and our own IPv6 address block. This prototype allows us to collect measurements that show the rich path diversity between edge networks, the differences in performance between paths and across time, and the value of collecting one-way measurements between edge networks. The prototype also shows how edge networks can readily adopt Tango.

In the next section, we discuss alternative approaches to improving wide-area performance, including multi-homed route control (at a single edge network), multipath TCP (on end hosts), and proposed multipath extensions to BGP. Each of these solutions has limitations in exposing path diversity, collecting performance measurements, or supporting incremental deployment. Then, Section 3 presents the Tango architecture and discuss how Tango overcomes these limitations. Next, in Section 4, we present our Vultr deployment of the Tango architecture, in which we conduct a measurement study that demonstrates the effectiveness of edge-network cooperation to expose useful path diversity. We summarize our comparison of one-way performance across paths in Section 5. Finally, Section 6 discusses promising research directions, including additional ways to expose wide-area path diversity, collect path measurements in an efficient and secure manner, and perform effective load balancing across multiple paths in the data plane.

2 MOTIVATION

Edge networks often have little visibility into, and control over, the wide-area paths that carry their traffic, leading to poorer and less predictable performance. Prior work on addressing these challenges faces limitations in measurement accuracy, path diversity, or incremental deployability.

2.1 Challenges in wide-area route control

To optimize routing, a network needs *visibility* (the ability to measure the alternative paths) and *control* (the ability to select among multiple paths). Both of these are hard today.

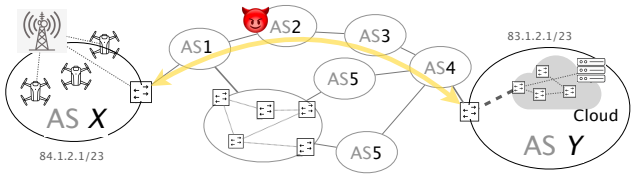

Figure 1: Status-quo: ASX and ASY are not multi-homed and have no infrastructure for accurate measurements; thus, they cannot measure, let alone optimize, the AS paths.

Inaccurate measurements of wide-area paths. Collecting wide-area performance measurements is particularly difficult today. First, end-to-end performance measurements are often dominated by problems in the edge network (*e.g.,* wireless interference or local congestion) or on the end-hosts themselves (*e.g.,* overloaded machine). Second, bidirectional metrics such as round-trip time (RTT)—whether collected by end-hosts or network devices—are hard to decompose into separate metrics for the two one-way paths. Third, while ECMP traffic splitting can increase the effective path diversity, it also complicates efforts to combine measurements from multiple flows that may traverse different paths. Finally, measurement strategies often rely on protocol semantics (*e.g.,* TCP sequence and acknowledgment numbers) or probing, which have multiple known disadvantages (*e.g.,* delayed acknowledgments). Such strategies do not generalize to all traffic *e.g.,* QUIC, or might be fooled by ASes treating probe traffic preferentially.

Insufficient path diversity for edge networks. Despite the rich path diversity of the Internet core, an edge network's interdomain routing choices are limited to its direct neighbors. A single-homed network has no choice at all, beyond using the one BGP route its one provider offers for each destination IP prefix. Plus, this route is likely suboptimal, as BGP does not make routing decisions based on performance metrics. A multi-homed network can select wide-area paths across its multiple providers. However, many networks have just two, or perhaps three, providers. Plus, these seemingly different wide-area paths might have common bottlenecks that make optimizing across them meaningless. Any source routing protocol or multipath extension to BGP would require the participation of multiple ASes, making deployment difficult.

2.2 Limitations of prior work

To highlight the challenges, consider the example in Figure 1 where ASX performs real-time analytics on drone data to enable adaptive control. To that end, ASX runs virtual machines (VMs) in a cost-effective and reliable cloud in ASY. Soon enough, ASX realizes that occasional increases in network delay hinder the drone applications. While ASY could deploy servers closer to ASX (edge-cloud), ASY cannot sustain the associated costs. Given the path diversity between ASX and ASY, and ample research on (performance-driven) route control [10, 18, 19, 30], ASX and ASY should be able to *(i) accurately* measure the performance of many of the paths;

(ii) direct traffic via the most performant paths in each direction; and *(iii)* do so without requiring support from the core or any third-party provider. A closer look shows that existing approaches cannot simultaneously offer accurate measurement, path diversity, and incremental deployability.

Multi-homing offers greater path diversity to stub ASes and has been heavily studied [18, 32**?**], including in the context of programmable data planes [10, 19]. However, this is not a viable solution for ASX and ASY as they are both single-homed. Even assuming one of them were multi-homed, the possible optimizations would be limited to one direction and to a small set of paths. Plus, each network's border switch would be limited to measuring traffic volumes and round-trip performance, rather than one-way performance metrics.

Multi-path TCP [15, 29] allows end-hosts to balance load over available paths, such as multiple network interface cards (*e.g.,* wireless and cellular). However, MP-TCP is difficult to translate into the interdomain setting. Indeed, connecting multiple AS-level paths to the end-host network stack is technically hard. Plus, end-host measurements are influenced by variable performance within the two edge networks. For example, the drones in ASX may experience link-layer retransmissions of corrupted packets in the wireless network, while the virtual machines in ASY may experience random delays in the hypervisor of the hosting servers.

Multipath extensions to BGP [13] could enable ASes to select and advertise multiple BGP paths for the same destination IP prefix. However, these modifications would need support from the ASes in the Internet core, where path diversity is available. In addition to extending the control plane, ASes would need support in the data plane to direct data packets over a specific path among the set of options. Furthermore, core ASes do not always have the same route-selection objectives as the edge networks; for example, core ASes often select paths based on business objectives rather than performance. Additionally, the BGP protocol does not expose any performance information about the underlying paths.

Overlay networks like RON [8] lay the foundations of optimized path selection over a virtual topology layered on top of the Internet. However, overlay networks require extra infrastructure (and associated costs), coordination and deployment challenges, and end-host overheads for software processing. More recent overlays from Akamai [5] and Cloudflare [22] rely on their own infrastructure and world-wide presence; these solutions do not apply to our example with a single access network and a datacenter.

New routing architectures such as SCION [24] have the potential to offer interdomain route control. Indeed, SCION supports multiple paths across the backbone that can be chosen by edge networks or end hosts. Unfortunately, SCION cannot meet our deployability requirements. First, SCION requires collaboration from both the core and edge networks. Second, running SCION incurs non-trivial costs associated with running the CPU-based packet processors as the protocol cannot be implemented on standard IP routers.

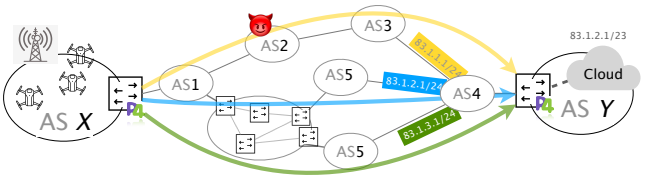

Figure 2: In Tango, ASX and ASY deploy programmable switches at their network borders to measure the wide-area paths and select specific paths via tunneling.

3 COOPERATIVE WIDE-AREA ROUTING

Tango's key insight is have edge networks *cooperate* to expose greater path diversity and collect more accurate measurements of path performance, without requiring changes to the core or introducing an extensive overlay network. Tango has three main components 1) a BGP interdomain control plane that is used to establish different paths between Tango switches, 2) a data plane implemented on programmable switches that attaches measurement data to packets and routes packets across the appropriate interdomain path, and 3) a local configuration containing the available routes to the other Tango switch and logic for how a forwarding decision should be made based on path performance. This setup allows us to overcome the core challenges of previous work.

Path diversity through cooperation. With edge-network cooperation, we overcome a fundamental limitation that has hindered interdomain route control: backbone routers only have a single forwarding decision installed for every destination IP prefix. This prevents the use of distinct routes for different applications or short-term route changes (that are too frequent and could overwhelm the control plane or too short-lived for BGP's several minute convergence time). Intradomain routing has overcome these concerns with different types of tunnelling technologies (*e.g.,* VLAN IDs, L3 encapsulation, and MPLS), but these are not supported across networks and require cooperation from the core.

Tango separates edge-network addressing from interdomain prefixes and leverages the fact that core routing tables already have multiple routes to a destination installed in the form of multiple IP prefixes. Previous work has focused on the idea of multiple routes to the same prefix (which fundamentally requires some type of change in the core), overlay routing (which eliminates the need for core participation but requires the operation of an entire overlay network), or single-hop route selection (where a multi-homed network selects its next hop but has no control of the route beyond that). Our work avoids this limitation by rethinking prefixes as *routes* (as opposed to groups of destinations) and allowing multiple prefixes representing different routes to reach the same destination. Enabling prefixes to propagate over specific routes is already well studied [12, 25] and is achievable with well established BGP techniques such as BGP communities [12, 27] and AS-path poisoning [25].

Even though each prefix is only associated with a single forwarding decision (as supported by current routers), each tango switch can be reached via multiple prefixes that it announces

over different interdomain paths. This allows for multi-path routing through the core without any cooperation and overcomes the deployment challenges associated with changing core routing (see Fig. 2 where all three prefixes can be used to reach AS *Y* over different paths). Furthermore, this configuration can be seen as a form of source routing on the backbone. The traffic source can select the prefix corresponding to its chosen route and pick the destination IP address so that traffic traverses its chosen path.

This type of configuration is uniquely enabled by our edge-network solution. While the destination edge network can advertise prefixes, cooperation from the source edge network is required to understand which prefixes are available to reach the destination and select which route to use on a per-packet basis.

To implement this technique, Tango switches announce multiple prefixes across different routes and then build tunnels with endpoints in those different prefixes. These tunnels traverse the different interdomain paths exposed by the different prefixes. Since prefixes in Tango are used to represent routes through the Internet, hosts need to use a different mechanism to specify locations on the Internet (*i.e.,* where their traffic is going). Tango enables this by having a distinct set of prefixes (not used for tunnels between Tango switches) that is used for host addressing (and can even be a different IP version). These host-address prefix(es) can be announced over traditional BGP to enable communication with non-Tango endpoints, but when the border router sees traffic destined for another Tango endpoint (based on a table which can be statically configured as both endpoints are cooperating), it makes a performance-driven/application-specific routing decision and forwards the traffic over the appropriate tunnel.

Accurate measurements through one vantage point at the edge of each AS. Tango leverages *switches* that are strategically placed at each network's edge. By doing so Tango can collect one-way measurements of the wide-area part of the path, *i.e.,* avoiding noise from the edge networks or end-hosts. Further, Tango tunnels traffic before forwarding it to each path to avoid unpredictable path diversity (*e.g.,* due to 5-tuple hashing in ECMP) which will result in measuring multiple paths as one. Finally, Tango uses existing data packets to piggyback measurement information on one side which the other can use and remove, effectively avoiding probing or protocol dependence.

To implement these measurements, Tango adds an IP tunnel header, a UDP header (to control ECMP behavior), and a timestamp to data packets. The destination switch records the timestamp and computes the difference between the timestamp and current system time before removing the encapsulation and forwarding the packet on to the end host. Even though the clocks may not be synchronized between the sending and receiving switches, all one-way delays calculated would be distorted by the same amount—still allowing for accurate *relative* comparisons of one-way delays.[1] Furthermore, adding tunnel-specific sequence numbers on packets can allow Tango to additionally compute loss and reordering.

[1]If Tango is implemented with more than one sending or receiving switch, all senders and receivers must have a form of relative clock synchronization to accurately compare measurements that go through different ingress/egress points.

4 PRELIMINARY PROTOTYPE

We deployed Tango between two datacenter edge networks —one in New York/New Jersey (NY) and one in Los Angeles (LA)— operated by the cloud provider Vultr. Our prototype consists of two servers (one at each datacenter) that were provided with auto-assigned IPv4 and IPv6 addresses and a single default gateway. We also utilized the free BGP connectivity Vultr provides to its tenants (which is offered to facilitate bring-your-own IP, and advanced interdomain networking like custom inbound traffic engineering, IP anycast, and smooth fail over between Vultr DCs [1]). To implement Tango in this environment we used the BIRD routing daemon [4] to provide Tango's BGP control plane, we wrote a data-plane implementation in eBPF (as we did not have access to programmable switches in Vultr's DCs), and we generated static configurations for tunnel endpoints using an IPv6 address block allocated to us by our affiliated institution. Although under the same authority, the two DCs communicate over the public Internet, with default routes through NTT's network.

4.1 Tango in the control plane.

The goal of the Tango control plane is to expose multiple AS-paths via which the two edge switches can forward traffic. This is challenging in our case because *(i)* we do not control a multi-homed AS, *(ii)* we do not have a perfect view of the AS topology between the two DCs, and *(iii)* Vultr does not own a private WAN.

To address these challenges we follow a three-step approach. The first step is to establish BGP sessions with Vultr's upstream router (this is not needed if an organization already uses BGP for its connectivity). The second step aims at identifying the available AS-paths from each server and finding the appropriate way to signal them using BGP. The third step aims at leveraging the exposed paths to dynamically route packets.

Step 1: propagate advertisements. We run a BIRD [4] instance on each of our cloud servers and set up an eBGP session between each server and the co-located Vultr router.[2] This allows our servers in each DC to propagate advertisements for IP prefixes (that we obtained from our affiliated academic institution) via Vultr's routers. In our setup, each server advertises four different /48 prefixes and uses BGP communities offered by Vultr to shape outbound BGP announcements [2].

Step 2: identify alternative paths. To explore the available paths between the NY server and the LA server we leverage our capability to propagate advertisements (as we explained before) and Vultr's support of standard traffic-control communities [2]. Prior research has shown that BGP communities are well supported across Internet providers [12]. In our context, BGP communities let us prevent export of our announcements

[2]To circumvent the lack of a legitimate ASN, these sessions were established with a private ASN that is removed from the AS path when Vultr propagates our advertisements to the Internet.

to select transit providers of Vultr. Leveraging this we implemented a simple but effective iterative algorithm of attaching a BGP community on a BGP advertisement that we send from one server and observing the AS-path heard at the other server. Concretely, to explore paths in a single direction between a source and a destination DC: 1) We observed the best BGP route for the destination exported by Vultr to our server at the source DC. 2) We configured our BIRD instance at the destination DC to attach a BGP community that would *suppress* this route. 3) We waited for BGP to propagate and confirmed that the source DC now sees an alternate route. 4) We recorded the communities and routes involved and repeated the process by measuring and adding an additional community. This was repeated until suppressing the used route caused the prefix to become unreachable by the other endpoint.

Following this procedure we found that the LA and the NY DCs are connected by at least four paths in each direction as we illustrate in Fig. 3. Specifically, traffic from LA to NY can be routed through (in order of preference by Vultr's routers): *(i)* NTT; *(ii)* Telia; *(iii)* GTT; and *(iv)* NTT and Cogent (we refer to this as Cogent). Traffic from NY to LA can be routed through: *(i)* NTT; *(ii)* Telia; *(iii)* GTT; and *(iv)* Level3. Moreover, we have recorded the communities and routes each server needs to propagate to expose them.

Step 3: forward traffic via a particular path Upon reception of a packet the Tango switch (server in our case) should be able to redirect it to its destination via one of the exposed paths in the previous step. However, BGP only uses a single path per destination. To address this, the Tango switch encapsulates the packet using an IP and UDP header that changes its destination IP such that it follows the chosen path.

4.2 Tango in the data plane

To goal of the Tango data plane is to *(i)* direct packets via the correct path by encapsulating them in an IP and UDP header and *(ii)* measure the one-way-delay of the different paths by piggy-packing information in the UDP header. Since we do not control a programmable switch, but want to reduce measurement noise and make our prototype scalable, we implement this functionality as a eBPF program. The sender-side eBPF program timestamps and encapsulates packets in a fixed IP and UDP header based on the chosen path for that packet. The receiver-side eBPF program calculates the difference between the current time and the timestamp to estimate the one-way delay. Each server runs both the sender and the receiver-side eBPF program. Observe that with two endpoints, all packets (regardless of path) are timestamped by a single sending machine and read by a single receiver, providing sound relative measurements between paths without assuming synced clocks (as the offset caused by out-of-sync clocks is a constant).

On the generality of our setup We stress that our set-up is well within the capabilities of small enterprises. First, we leverage features that Vultr offers to all clients for free. Second, the use of an IPv6 prefix is not hard, considering *(i)* the abundance in IPv6 space and *(ii)* cloud providers also lease space *e.g.,* Vultr.

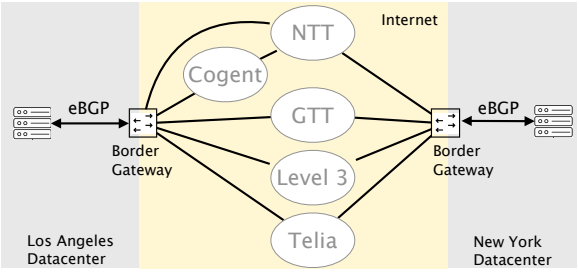

Figure 3: Multiple paths used between Vultr DCs.

5 PRELIMINARY RESULTS

Our set-up shows that a Tango infrastructure is practical to instantiate even with minimal resources. In this section, we investigate the expected gain from using the Tango. While our measurements do not translate to every single case in the Internet, it does show that even short AS-paths that traverse the core might not be optimal. We first explain our methodology before we describe our results.

Measurement Methodology We ran the eBPF program in our two servers for an eight-day period and recorded the average one-way delay for every path at 10ms intervals. To generate traffic we ran a ping along each path every 10ms. Because of the lack of clock sync, comparisons between one-way-delays in different directions have little meaning, but relative changes in one-way-delay between paths are meaningful because the clock offset is a constant.

The BGP default path is 30% worse than the most performant path. This result highlights the potential benefit of Tango even in seemingly short paths as in our prototype. We illustrate our results of the one-way delay over time across the different paths between NY and LA in Fig. 4. GTT's path significantly outperforms the BGP default path through NTT whose delay is 30% higher on average. The same holds for the reverse direction. In addition, we see that different paths show different jitter characteristics. To measure sub-second network jitter, we calculated the mean standard deviation of a 1-second rolling window. For example, in the LA to NY direction (not shown in the figure) we found the least noisy path GTT had a rolling window standard deviation of .01ms while Telia had a deviation of .33ms. Depending on the application, delay and jitter could have a significant impact on the user-perceived performance. While we observe some correlated disturbances, each path exposes unique performance characteristics. In conclusion, even a tenant in Vultr could benefit from Tango today.

Our measurement methodology allows us to monitor the delay of the exposed paths over a longer time period. Doing so allows us to find temporal performance changes worth being realized or avoided with adaptive routing. Next, we explain two particularly interesting incidents that make the case for continuous measurements and dynamic route control.

Internal routing changes Fig. 4 shows an hour-long frame of our experiment. Around hour 121.25, the one-way-delay of GTT's route dramatically increases during a brief period of

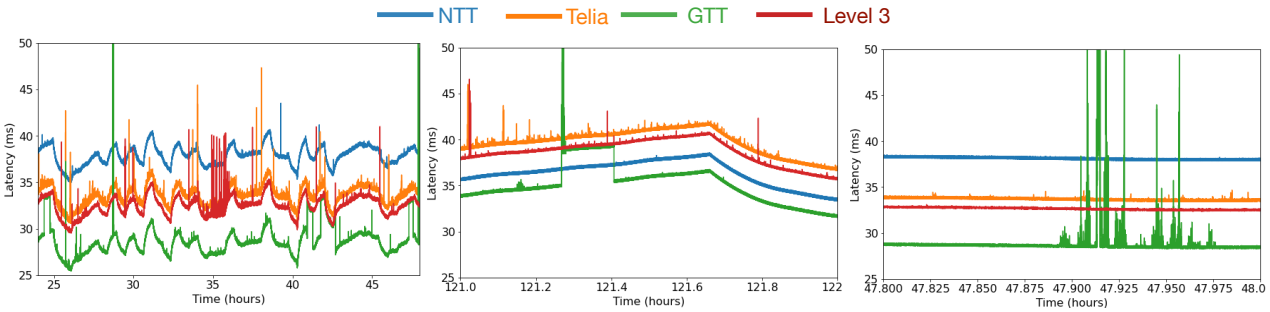

Figure 4: One-way-delay of different paths from NY to LA (left) showing the best path (GTT in green) outperforming the default path (NTT in blue). A route change (middle) and a period of instability (right) with latency spikes up to 78ms in GTT's network in the NY to LA direction.

instability. After this, it quickly stabilizes at a new minimum that has a 5ms longer one-way delay. This persists for around 10 minutes until the original path is used. Thus, during these route-change events, selecting an alternate path based on live data is required for optimal performance.

Periods of network instability At various points in the data the networks seem to undergo a period of poor performance. Several of these times are visible in the 24h trace in Fig. 4. Fig. 4 also shows a close-up of one such event. The period of instability lasts approximately 5min and involves both minor increases in one-way delay and major spikes resulting in a peak one-way-delay of 78ms (more than double the minimum one-way delay of 28ms). During this time, all other networks experience almost no interference to their usual operations maintaining their minimum one-way delays. During these spikes, a latency-sensitive application (e.g., drone-control) could experience significantly degraded performance. Furthermore, even though GTT's network does deliver some packets at the minimum one-way delay of 28ms (even during the instability), TCP's in-order packet delivery means that should a packet experience delay during one of these spikes, future application packets will be delivered out-of-order (resulting in a reduction in TCP throughput) and the application-layer data stream will be held up by the slow packet. Thus, changing to a path that is not experiencing this network instability is superior for application performance.

6 FUTURE RESEARCH

Support for wide-area, efficient & trustworthy telemetry. Any data-driven system working in the wide-area is vulnerable to on-path and off-path attackers who might try to compromise the monitoring process. For instance, an attacker might try to inject, drop or modify some of the packets used for measurements. In theory, the two Tango end-points can use cryptography to protect the process.While several pieces of existing work have tried to minimize the potential impact of an on-path adversary to the measurements [11, 16, 17, 23], none of those facilitates the exchange of arbitrary measurement information or is made to work under the resource constraints of typical programmable switches.

From Tango of 2 to Tango of N. In this paper we focus on the opportunities that arise when two parties collaborate under the Tango architecture using one point of presence (PoP) each. This is a crucial step as it allows Tango to be incrementally deployable at no extra cost and worths being automated using more knobs such as AS-path poisoning, ECMP reverse engineering *etc.* Still, a Tango of two only barely scratches the surface of Tango's full potential. We envision, Tango of two to be the building block of an open and robust wide-area overlay composed of more networks and of more PoPs of the same network. Doing so will expose a larger path diversity to Tango participants using RON-like techniques.

Tango to make complex wide-area measurements and optimization practical today. Over the years multiple measurement and route optimizations techniques with different targets and trade-offs have been proposed [6, 7, 9, 14, 20, 21, 26, 28, 31]. Their wide-area deployment though is hindered by the lack of a robust overlay that is BGP-compliant, cost-efficient and does not require special hardware. We believe that realizing Tango will trigger the re-evaluation of such techniques. Finally, Tango has the potential to act as a wide-area dynamically slicable network allowing participants to enforce certain QoS.

ACKNOWLEDGMENTS

We would like to thank the Princeton University Office of Information Technology for allowing us to announce part of Princeton's IPv6 allocation for research purposes. We are also thankful for support from DARPA under grant HR001120C0107.

REFERENCES

[1] 2022. Announce your IP Space with BGP and Vultr - Vultr.com. https://www.vultr.com/features/bgp/. (2022).
[2] 2022. AS20473 BGP Customer Guide. https://www.vultr.com/docs/as20473-bgp-customer-guide. (2022).
[3] 2022. SSD VPS Servers, Cloud Servers and Cloud Hosting. https://www.vultr.com/. (2022).

[4] 2022. The BIRD Internet Routing Daemon Project. https://bird.network.cz/. (2022).

[5] Akamai. 2022. SureRoute. https://developer.akamai.com/article/sureroute. (2022).

[6] Aditya Akella, Bruce Maggs, Srinivasan Seshan, and Anees Shaikh. 2008. On the performance benefits of multihoming route control. *IEEE/ACM Transactions on Networking (TON)* 16, 1 (2008), 91–104.

[7] Aditya Akella, Srinivasan Seshan, and Anees Shaikh. 2004. Multihoming Performance Benefits: An Experimental Evaluation of Practical Enterprise Strategies.. In *USENIX Annual Technical Conference, General Track*. 113–126.

[8] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. 2001. Resilient Overlay Networks. In *ACM Symposium on Operating Systems Principles (SOSP '01)*. 131–145. https://doi.org/10.1145/502034.502048

[9] Maria Apostolaki, Gian Marti, Jan Müller, and Laurent Vanbever. 2019. SABRE: Protecting Bitcoin against Routing Attacks. In *Network and Distributed System Security Symposium (NDSS)*.

[10] Maria Apostolaki, Ankit Singla, and Laurent Vanbever. 2021. Performance-Driven Internet Path Selection. In *ACM SIGCOMM Symposium on SDN Research (SOSR)*. 41–53. https://doi.org/10.1145/3482898.3483366

[11] Ioannis Avramopoulos and Jennifer Rexford. 2006. Stealth Probing: Efficient Data-Plane Security for IP Routing. In *USENIX Annual Technical Conference*. USENIX Association, Boston, MA. https://www.usenix.org/conference/2006-usenix-annual-technical-conference/stealth-probing-efficient-data-plane-security-ip

[12] Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. 2019. SICO: Surgical Interception Attacks by Manipulating BGP Communities. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 18. https://doi.org/10.1145/3319535.3363197

[13] Jose M Camacho, Alberto García-Martínez, Marcelo Bagnulo, and Francisco Valera. 2013. BGP-XM: BGP Extended Multipath for Transit Autonomous Systems. *Computer Networks* 57, 4 (2013), 954–975.

[14] A. Elwalid, C. Jin, S. Low, and I. Widjaja. 2001. MATE: MPLS adaptive traffic engineering. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, Vol. 3. 1300–1309 vol.3. https://doi.org/10.1109/INFCOM.2001.916625

[15] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch. 2020. *TCP Extensions for Multipath Operation with Multiple Addresses*. RFC 8684. RFC Editor.

[16] Sharon Goldberg and Jennifer Rexford. 2007. Security vulnerabilities and solutions for packet sampling. In *IEEE Sarnoff Symposium*. 1–7. https://doi.org/10.1109/SARNOF.2007.4567339

[17] Sharon Goldberg, David Xiao, Eran Tromer, Boaz Barak, and Jennifer Rexford. 2008. Path-Quality Monitoring in the Presence of Adversaries. In *ACM SIGMETRICS*. Association for Computing Machinery, New York, NY, USA, 193–204. https://doi.org/10.1145/1375457.1375480

[18] David K Goldenberg, Lili Qiu, Haiyong Xie, Yang Richard Yang, and Yin Zhang. 2004. Optimizing cost and performance for multihoming. In *ACM SIGCOMM*, Vol. 34. ACM, 79–92.

[19] Thomas Holterbach, Edgar Costa Molero, Maria Apostolaki, Alberto Dainotti, Stefano Vissicchio, and Laurent Vanbever. 2019. Blink: Fast connectivity recovery entirely in the data plane. In *USENIX Symposium on Networked Systems Design and Implementation*. 161–176.

[20] Srikanth Kandula, Dina Katabi, Bruce Davie, and Anna Charny. 2005. Walking the tightrope: Responsive yet stable traffic engineering. In *ACM SIGCOMM Computer Communication Review*, Vol. 35. ACM, 253–264.

[21] Changhoon Kim, Anirudh Sivaraman, Naga Praveen Katta, Antonin Bas, Advait Dixit, and Lawrence J Wobker. 2015. In-band Network Telemetry via Programmable Dataplanes *(Industrial demo, ACM SIGCOMM '15)*.

[22] Rustam Lalkaka. 2019. Argo and the Cloudflare Global Private Backbone. (Dec 2019). https://blog.cloudflare.com/argo-and-the-cloudflare-global-private-backbone/.

[23] Myungjin Lee, Sharon Goldberg, Ramana Rao Kompella, and George Varghese. 2014. FineComb: Measuring Microscopic Latency and Loss in the Presence of Reordering. *IEEE/ACM Transactions on Networking* 22, 4 (2014), 1136–1149. https://doi.org/10.1109/TNET.2013.2272080

[24] Adrian Perrig, Pawel Szalachowski, Raphael M. Reischuk, and Laurent Chuat. 2017. *SCION: A Secure Internet Architecture*. Springer Verlag.

[25] Alex Pilosov and Tony Kapela. 2008. Stealing the Internet: An Internet-scale man in the middle attack. *NANOG-44, Los Angeles, October* (2008), 12–15.

[26] Stefan Savage, Thomas Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoff Voelker, et al. 1999. Detour: Informed Internet routing and transport. *IEEE Micro* 19, 1 (1999), 50–59.

[27] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. BGP Communities: Even More Worms in the Routing Can. In *ACM Internet Measurement Conference (IMC '18)*. 279–292. https://doi.org/10.1145/3278532.3278557

[28] Hongsuda Tangmunarunkit, Ramesh Govindan, and Scott Shenker. 2001. Internet path inflation due to policy routing. In *ITCom 2001: International Symposium on the Convergence of IT and Communications*. International Society for Optics and Photonics, 188–195.

[29] Damon Wischik, Costin Raiciu, Adam Greenhalgh, and Mark Handley. 2011. Design, implementation and evaluation of congestion control for multipath TCP. In *USENIX Networked Systems Design and Implementation*.

[30] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew Holliman, Gary Baldus, Marcus Hines, Taeeun Kim, Ashok Narayanan, Ankur Jain, Victor Lin, Colin Rice, Brian Rogan, Arjun Singh, Bert Tanaka, Manish Verma, Puneet Sood, Mukarram Tariq, Matt Tierney, Dzevad Trumic, Vytautas Valancius, Calvin Ying, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. 2017. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 432–445. https://doi.org/10.1145/3098822.3098854

[31] Li Yuliang, Miao Rui, Kim‡ Changhoon, and Yu Minlan. 2016. LossRadar: Fast Detection of Lost Packets in Data Center Networks. In *CoNEXT*. ACM, New York, NY, USA, 15.

[32] Zheng Zhang, Ming Zhang, Albert Greenberg, Y. Charlie Hu, Ratul Mahajan, and Blaine Christian. 2010. Optimizing Cost and Performance in Online Service Provider Networks. In *USENIX Networked Systems Design and Implementation*.