Poster: Voltadomar : A `Traceroute` Library for Anycasted IPs

David Song
University of Washington
Seattle, WA, USA

Innocent Ndubuisi-Obi Jr
University of Washington
Seattle, WA, USA

Kurtis Heimerl
University of Washington
Seattle, WA, USA

ACM Reference Format:
David Song, Innocent Ndubuisi-Obi Jr, and Kurtis Heimerl. 2025. Poster: Voltadomar : A `Traceroute` Library for Anycasted IPs . In *Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25), October 28–31, 2025, Madison, WI, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3730567.3768602

1 Introduction

Prior work has aimed to characterize [1] or geo-locate [2] IP anycast by running probes from unicast *to* anycast prefixes. These works have either used `ping` *as-is* on unicast clients or adapted it to work from anycast clients [3, 4]. Our research question is related to the inverse: how can we understand paths *from* anycast *to* unicast prefixes? Answering this question requires that we adapt `traceroute` to work from anycast clients. As demonstrated in Fig. 1, when anycasted **Node A** sent `traceroute` probes 1 and 2 towards a destination on the Internet, its `traceroute` process never received the ICMP replies which were instead forwarded to **Node B**. This is because `traceroute` breaks when used *as-is* from anycast infrastructure: *ICMP replies may be directed toward a different anycasted node than the sender, breaking the path reconstruction required by* `traceroute`. In order to capture the full potential of using anycast research infrastructure to understand the emerging dynamics on today's Internet, we need to support running traceroutes from anycasted nodes.

2 System Design

To resolve this issue, we design and implement a new measurement tool and system called **voltadomar**. Voltadomar is a Python library and set of component interfaces for issuing traceroutes from anycasted vantage points. Our library is oriented around three core design decisions informed by our review of the original traceroute implementation by Van Jacobson and the ICMP RFC 792: (1) separating mechanism and policy, (2) inverting control, and (3) supporting unique, identifiable, and authenticated probes.

Separating Mechanism and Policy. As previously discussed, vanilla traceroute on anycast is unable to reconstruct paths when replies are routed to a different anycasted node. In voltadomar, we solve this by separating traceroute policy from the mechanism: we move state related to *path reconstruction* to a dedicated component, the `Controller`, which is connected to the voltadomar `Agent` running on each anycasted node (See Fig. 2). The remaining mechanism of sending of probes is left to the `Agent` running on each anycast node.

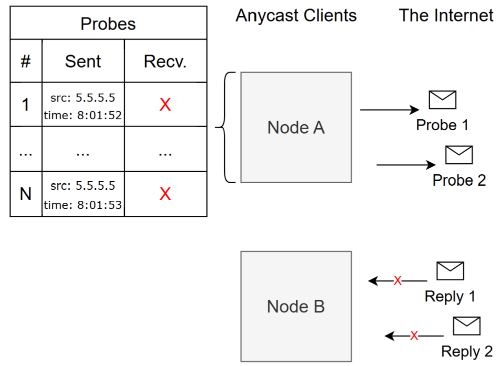


Figure 1: Pitfalls of vanilla traceroute on anycast

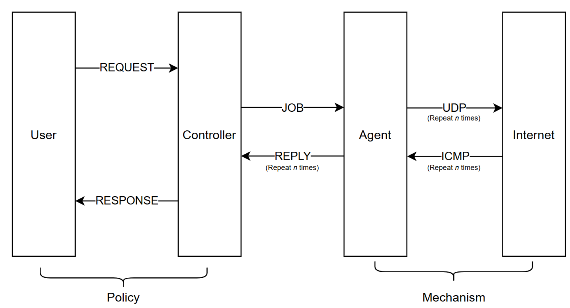


Figure 2: Voltadomar's approach to traceroute

Each `Agent` also forwards all received replies to the `Controller`. The `Controller` is responsible for aggregating path reconstruction state from `Agents`. By separating the policy (the traceroute path reconstruction state and logic) from the mechanism (sending probes/forwarding replies), we can support running traceroute across a set of anycasted vantage points. Further, this separation allows us to support different implementations of traceroute (e.g. Paris traceroute, TCP, etc) because policy logic is centralized at the `Controller` while minimal to no modifications are needed in the mechanism.

Inversion of Control. Drawing on prior work demonstrating the efficacy of inverting network measurements for anycast networks [3], our methodology implements a inverse-probing technique: we encapsulate traceroute requests from a user and send them to the `Controller`, which then coordinates requests to `Agents` and gathers results on behalf of the user. This inversion allows us to leverage to *separation of mechanims and policy*: we forward ICMP replies from the vantage points to voltadomar traceroute

IMC '25, Madison, WI, USA

ACM ISBN 979-8-4007-1860-1/2025/10
https://doi.org/10.1145/3730567.3768602

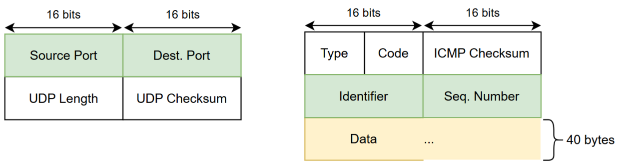
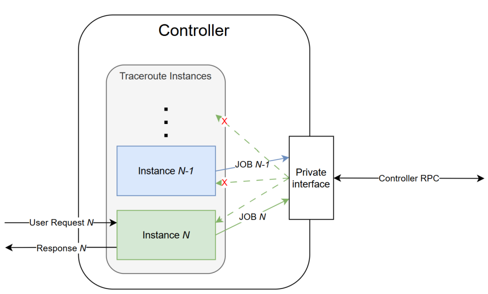


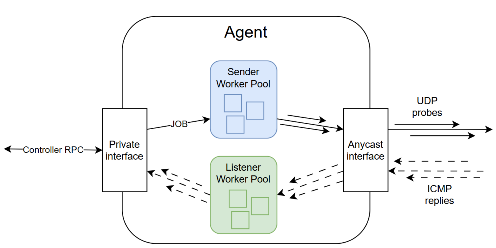
Figure 3: UDP header (left) vs. ICMP Echo header (right)

instances running on the Controller who aggregates and sends the reconstructed paths to the user (See Figure 4a).

Unique, Identifiable, and Authenticated Probing. Finally, to (re)identify probes in order to reconstruct paths at the `Controller`, we require a method of encoding the unique identifier for a probe. The unique identifier must distinguish (1) the vantage point that sent the probe, (2) the sequence number for the probe, and (3) the traceroute session at the controller that requires the probe's result. Normally, vanilla traceroute uses two main probing mechanisms: ICMP or UDP. With ICMP, it sends an ICMP Echo and listens for the corresponding Reply message. With UDP, it sends empty UDP datagrams and listens for an Time Exceeded or Destination Unreachable message. Both protocols provide 32 bits in the header which we can use to identify probes. ICMP has an additional 40-byte payload, but, in practice, we found this payload is unreliable. When conducting ICMP traceroute measurements over our University network to a set of 16 popular DNS service providers, we found that >40% of routers dropped the 40-byte payload. As a result, we can only consistently rely on the 32 bits in the header (UDP and ICMP) for probe identification and authentication.



(a) Controller creates a traceroute instance per use request which associates packets and stores state



(b) Agent leverages two worker pools to dispatch jobs and forward ICMP replies to the controller

3 Experimentation

As a proof of concept, we tested voltadomar on a simple virtual network testbed to simulate an anycast setting. The network consisted of four virtual machines (VMs) representing two anycast nodes, the controller/client interface, and a target. To simulate anycast routing, we periodically manually modified the target VMs ARP table to toggle between the two anycast node MAC addresses. Voltadomar was able to successfully re-associate replies which were received by different anycast nodes.

4 Future Work

In our future work, we hope to expand this work by: (1) implementing an HMAC in packet header to authenticate ICMP responses, (2) supporting all vanilla traceroute option flags, (3) deploying voltadomar in an (live) anycast network, and (4) supporting more network measurement tools with voltadomar including, but not limited to, Paris traceroute, ping, etc.

References

[1] Rui-Ling Bian, Shuai Hao, Haining Wang, Amogh Dhamdere, Alberto Dainotti, and Chase Cotton. 2019. Towards passive analysis of anycast in global routing: unintended impact of remote peering. *Comput. Commun. Rev.* 49 (2019), 18–25.

[2] Danilo Cicalese, Diana Joumblatt, Dario Rossi, Marc-Olivier Buob, Jordan Augé, and Timur Friedman. 2015. A fistful of pings: Accurate and lightweight anycast enumeration and geolocation. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2776–2784. doi:10.1109/INFOCOM.2015.7218670

[3] Wouter B. de Vries, Ricardo de Oliveira Schmidt, Wes Hardaker, John S. Heidemann, P. T. de Boer, and Aiko Pras. 2017. Broad and load-aware anycast mapping with verfploeter. *Proceedings of the 2017 Internet Measurement Conference* (2017).

[4] Raffaele Sommese, Leandro Marcio Bertholdo, Gautam Akiwate, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, Kimberly C. Claffy, and Anna Sperotto. 2020. MAnycast2: Using Anycast to Measure Anycast. *Proceedings of the ACM Internet Measurement Conference* (2020).