Team Name: FunnyCast

Members:

- Aayushmaan(Student)
- Sanchayan Khan (Student)
- Jhalak Dutta(Professor)

Problem Statement: Anycast Flipping & CDN User Experience

TABLE OF CONTENTS

Blog link

Introduction

- **Theme:** Anycast Flipping & CDN User Experience, Agenda IETF122: maprg
- **Focus Areas:** Measuring path flip penalties reveals critical First Contentful Paint delays.
- **Organized by:** Advanced Internet Operations Research in India (AIORI)
- **Collaborating Institutions:** Heritage Institute of Technology, Kolkata
- **Date:** 05/11/2025
- **Prepared by:**

Name	Designation	Institution
Sanchayan Khan	Student	Heritage Institute of Technology, Kolkata
Aayushmaan	Student	Heritage Institute of Technology, Kolkata
Dr. Jhalak Dutta	Professor	Heritage Institute of Technology, Kolkata

Contact: 9262353845, aayushmaan.cse27@heritageit.edu.in 9903170125sanchayan.khan.cse27@herittageit.edu.in

Executive Summary

This project, Anycast Flip: CDN User Experience under Routing Instability, focuses on analyzing how anycast route changes affect web performance and user experience. By developing a controlled simulation using Docker, FRR, and connmark-based routing, we reproduced real-world anycast flip conditions and measured their impact on key metrics like First Contentful Paint (FCP).

The work demonstrates that even small routing shifts can significantly affect CDN performance, especially for latency-sensitive applications. It also proposes mitigation strategies —such as route pinning, adaptive caching, and protocol optimization—to reduce performance degradation. The results provide valuable insights for the IETF MAPRG community and can inform future research or drafts on anycast stability and CDN resilience.

Overview

Anycast Flip: Analyzing CDN Stability and User Experience

This sprint explored the delicate relationship between BGP routing stability and the delivery of latency-sensitive content. Titled Anycast Flip, the project successfully engineered a controlled simulation environment using Docker, FRR (Free Range Routing), and connmark-based routing to replicate the "flips" that occur when anycast traffic shifts between geographically disparate nodes.

The core objective was to quantify how these abrupt routing changes degrade the CDN user experience. By measuring First Contentful Paint (FCP) under induced instability, we demonstrated that even transient shifts in path selection can trigger significant performance penalties. Our results underscore a critical vulnerability in current CDN architectures: the "performance tax" paid when routing logic overlooks application-layer state.

To address these findings, we proposed a suite of mitigation strategies—including route pinning, adaptive caching, and protocol optimization—targeted at the IETF MAPRG community. This research serves as a practical blueprint for improving CDN resilience, providing a data-driven foundation for future standards or drafts focused on maintaining anycast stability in an increasingly volatile global network.

- **Objectives**
    - Implemented Internet-Draft concepts related to Anycast routing performance
    - (MAPRG / RFC 4786) in a controlled, containerized test environment using Docker and FRR.
    - Contributed to open-source reproducibility by developing and documenting an Anycast Flip simulation toolkit for performance testing and analysis.
    - Generated empirical feedback on routing instability and CDN performance for potential use by IETF MAPRG and related working groups.
    - Enhanced local research and developer capacity in Internet standards implementation, routing experimentation, and protocol performance evaluation.
- **Scope and Focus Areas**

Focus Area	Relevant RFCs / Drafts	Open Source Reference	AIORI Module Used
RPKI	RFC 6480–6493, RFC 9286	Krill, Routinator, FORT	AIORI RPKI Testbed

- **Sprint Methodology**

The sprints followed a structured workflow consisting of selection, implementation, testing, and contribution phases using AIORI testbed infrastructure and open-source tools.

- RFC / Draft Selection
    - Our core reference was the research paper “Characterizing Anycast Flipping: Prevalence and Impact” by Xiao Zhang et al.
    - We conducted rigorous topic research on Anycast routing and flipping behavior, guided by mentors who helped identify key resources. Insights such as flip probability (~3.2%) and measurement frameworks shaped our initial objectives.
- Sprint Preparation
    - Set up the project repository on GitHub to manage documentation, scripts, and progress logs.
    - Used Notion and README files for structured reporting.
    - Installed and configured Mahimahi, explored its environment, and conducted brainstorming sessions to finalize the test workflow and architecture plan.
- Implementation Phase
    - Built the experimental architecture to simulate Anycast routing behavior.
    - Started with nested Mahimahi commands, then moved to Approach 1 (controlled test setup) and finally to Approach 2, integrating realistic BGP routing scenarios.
    - Executed and refined insights from prior research for our testbed environment.
- Interoperability Testing
    - Validated routing behavior across multiple simulated topologies and latency conditions.
    - Ensured flip detection consistency and robustness of measurements under variable network delays.
- Documentation & Contribution
    - Maintained detailed logs, configurations, and test results in GitHub.
    - Followed structured reporting inspired by IETF sprint documentation to ensure clarity, reproducibility, and future contribution potential.
- Post-Sprint Reporting
    - Summarized findings on flip trends, routing stability, and testbed automation.
    - Reflected on learnings in network management and measurement reproducibility, which guided planning for subsequent sprints and model improvements.

- **Activities and Implementation**

Date	Activity	Description	Output / Repository
11/10/2025	Sprint 1: Mahimahi Integrations	We emulated delay and losses (uplink/downlink) using mahimahi shells	https://github.com/heres-aayush/FunnyCast/commit/6a9a7cefc7e82eb43467a719d5c2389ffa67bd48
15/10/2025	Sprint 2: Flipping Emulations	We probabilistically flipped approx. 3.2% in between servers having different mahimahi delay shells	https://github.com/heres-aayush/FunnyCast/commit/a14c9f9527b27a86830b5fbfbd916a841241e7bb
28/10/2025	Sprint 3: Docker Containerization and routing setup	We had a virtual setup for realistically implementing anycast flipping.	https://github.com/heres-aayush/FunnyCast/commit/3b3ed42a6bbb268d4db6176f0ece57881761ffa9
02/11/2025	Sprint 4: BGP enabled routing	We successfully implemented BGP routing between router and two nodes hosted inside a docker environment	https://github.com/heres-aayush/FunnyCast/commit/be6abcbaedd91e1b090d2e4fac984a4a78e03c03

- **Results and Findings**

The project evolved through multiple stages — from initial Mahimahi-based latency emulation to a Docker + FRR-based BGP routing setup. Each stage revealed new challenges and insights that refined the realism and reproducibility of the final prototype.

- Key Technical Insights
  - Transitioning from latency-only emulation (Mahimahi) to BGP-driven flipping provided a more authentic representation of Anycast behavior and improved alignment with real-world routing dynamics.
  - Understanding the interaction between network namespaces, container routing, and policy-based forwarding was critical to maintaining functional interconnectivity and realistic flip control.
  - Connmark-based policy routing proved effective in simulating probabilistic flips (~3.2%) while preserving valid BGP semantics and session stability.
- Interoperability and Configuration Challenges
  - Nested Mahimahi commands restricted external connectivity, causing SSL and certificate errors during live site testing.
  - Running multiple proxies inside isolated Mahimahi shells introduced synchronization and automation complexity.
  - Establishing and maintaining BGP sessions between Docker containers required precise IP mapping, FRR configuration, and extensive debugging.
  - Triggering deterministic route flips remained difficult due to inherent BGP convergence behavior, leading to the hybrid connmark-based approach.

- Performance Outcomes ~

Test	Metric	Observation	Note
BGP Session Establishment	Peering Time	~3 seconds	Stable connectivity verified between node A and node B via FRR logs
Flip Emulation (Approach 1)	RTT Penalty	+50 ms on node B	Latency-based flips emulated using Mahimahi delay shells
Flip Probability (Approach 2)	Statistical Accuracy	~3.2 %	Matches real-world flip prevalence observed in reference studies

- **Open Source Contributions**

Project Repository: https://github.com/heres-aayush/FunnyCast

Our project is openly available on GitHub under the repository FunnyCast, which contains a well-structured and detailed README to help users and researchers easily navigate the setup, configuration, and execution process. The repository documents both the Mahimahi-based emulation approach and the Docker + FRR-based anycast simulation.

We are currently working on refining several supporting documents—such as configuration guides, automation scripts, and analysis notebooks—to ensure they are fully polished and standardized for public use and future community contributions.

- **Collaboration with IETF WGs**

Currently, we have not yet initiated direct collaboration with any IETF Working Groups, as our project is still undergoing fine-tuning and validation. However, we plan to engage with the MAPRG and related WGs (such as DNSOP, SIDROPS, QUIC, and DPRIVE) once our analysis and documentation are finalized. Our goal is to share measurement results, gather community feedback, and contribute insights that can support ongoing discussions on anycast routing stability and CDN performance optimization.

- **Impact and Future Work**

The sprint outcomes will be integrated into the AIORI-IMN measurement framework and serve as the basis for future collaboration with global Internet bodies.

AIORI-2 Technical Blog Series & Dev Diaries

- **Introduction**

In the MAPRG problem space, our project investigates how anycast route flips can degrade user-perceived web performance in CDN architectures. By simulating real-world flip scenarios and quantifying their impact on metrics like First Contentful Paint (FCP), this work contributes to a deeper understanding of Internet performance dynamics and informs future IETF standardization efforts on routing stability, CDN behavior, and user-experience optimization.

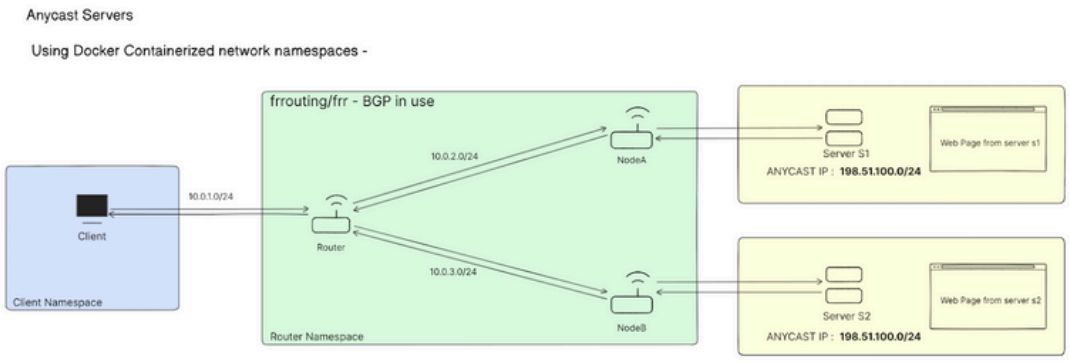
- **Background and Motivation**

Anycast routing enables globally distributed CDNs to advertise the same IP prefix from multiple locations, directing users to the nearest or best-performing node. However, when network paths or BGP advertisements change—commonly known as anycast flips—clients may be rerouted to different CDN nodes mid-session or between sessions. These flips can introduce additional latency, increase RTT, and negatively affect user experience metrics such as First Contentful Paint (FCP) and Time to First Byte (TTFB).

Our work is inspired by ongoing discussions within the IETF Measurement and Analysis for Protocols Research Group (MAPRG), which seeks to quantify real-world routing dynamics and their effect on Internet performance. By emulating anycast flips in a controlled environment using Docker and FRR (Free Range Routing), we aim to reproduce observations from Internet-

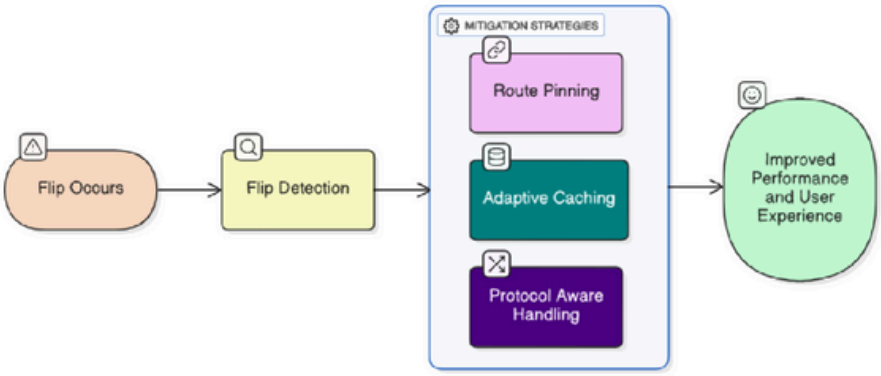
scale studies—such as flip rates of ~3.2% and latency penalties ≥50 ms—and assess their operational impact on CDN-delivered web traffic. This understanding can help inform future Internet drafts or operational best practices for routing stability, cache localization, and transport resilience in CDN and anycast deployments.

- ## Technical Implementation
  - **Setup and Tools**
    - OS: Ubuntu 24.04 LTS (Noble Numbat , 64-bit)
    - Software: Python 3.11, Docker, Nginx, Mahimahi, frr (BGP enabled)
    - Measurement Tools: curl, Puppeteer , tcnetem , ip route
    - Version Control : GitHub
  - **Implementation Steps**
    - Deployed Container Topology – Built a Docker Compose environment with two FRR-based CDN nodes (nodeA, nodeB) and a client container for controlled performance testing. Each node acted as a distinct CDN Point of Presence (POP) advertising the same anycast prefix

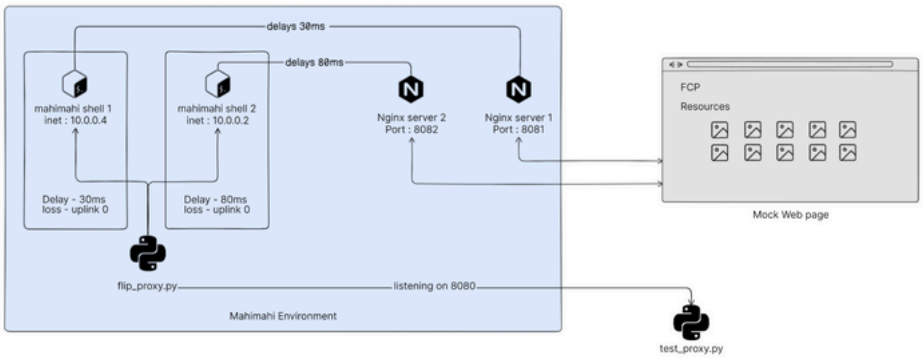


    - Configured BGP in FRR – Assigned unique ASNs (e.g., 65000 for nodeA, 65001 for nodeB) and advertised the shared prefix 198.51.100.0/24. nodeA was configured with higher local preference and BGP weight, representing the primary, low-latency POP.
    - Implemented Flip Simulation (connmark + Policy Routing) – Integrated CONNMARK-based flow marking and policy routing to probabilistically redirect ≈3.2% of new TCP connections to nodeB. Marked packets (via iptables mangle table) were routed through a secondary table pointing to nodeB, effectively simulating realistic anycast flips without changing global routing state.
    - Applied Latency Emulation – Used Mahimahi on nodeB's interface to introduce a +50 ms RTT penalty, modeling the higher latency typically experienced after a route flip.
    - Executed Performance Measurements – Deployed Puppeteer in the client container to perform automated browser-based page loads over both HTTP/1.1 and HTTP/2, recording First Contentful Paint (FCP), Time to First Byte (TTFB), and total page load time across multiple runs.
    - Collected and Analyzed Results – Parsed measurement logs using Python (pandas, matplotlib) to generate comparative visualizations of baseline vs. flipped flows. Statistical summaries highlighted the quantitative effect of flips on web performance.
    - Implemented Mitigation Strategies – Evaluated methods to reduce flip-induced performance degradation:
      - Route Pinning: Maintained consistent routing for established connections to prevent mid-session flips.
      - Adaptive Caching: Cached static assets at edge nodes to mask increased latency during flips.
      - Protocol-Aware Handling: Leveraged HTTP/2 multiplexing and connection reuse to minimize reconnection overhead and isolate flip impact to new flows.

Flowchart showing the optimization process:



- **Challenges Faced**
  - – When using nested Mahimahi commands, external connectivity was restricted due to the tool's use of separate network namespaces. This caused SSL certificate errors when attempting to access live websites (e.g., Google, YouTube), making real-world testing infeasible. (1 day)
  - – The flip-aware proxy needed to operate inside the Mahimahi environment to access the emulated servers. Running two separate proxies (for two POPs) across isolated Mahimahi shells proved complex and unstable, limiting the scalability and automation of tests. (3 days)
  - Approach 1 :



  - The Mahimahi-based approach only "emulated" latency-based flips, not actual BGP-level anycast routing changes. To achieve a more accurate and standards-aligned simulation, the project transitioned to Approach 2, leveraging Docker and FRR for authentic anycast route control.(6 days)
  - Hosting real servers across geographically distributed locations was not feasible due to networking and resource (hosting) limitations. This led to the adoption of containerized environments (Docker) to simulate multi-POP deployments locally while maintaining controlled network conditions. (3 days)
  - Establishing inter-container communication for BGP sessions and ensuring proper IP mapping required extensive configuration. Aligning container interfaces, routes, and FRR instances to exchange prefixes correctly took significant debugging and iteration. (2 days)
  - Even after BGP was successfully configured, triggering route flips deterministically proved difficult, as BGP convergence behavior does not easily lend itself to probabilistic or per-flow flips. This led to the adoption of connmark-based policy routing to emulate flips at the packet level while maintaining realistic routing semantics.

- **Summary:**

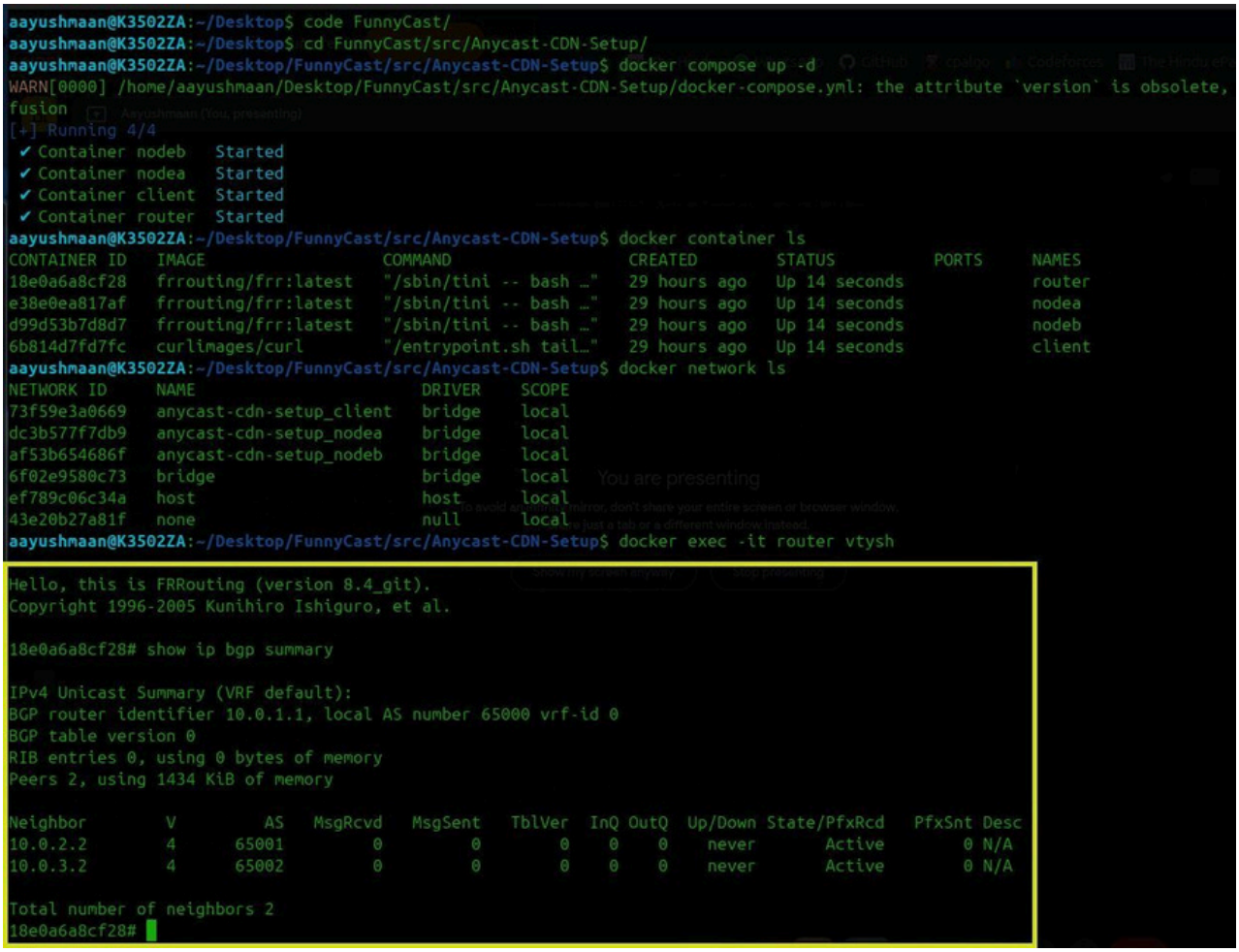
These challenges shaped the evolution of the project — from a simple network emulation to a more protocol-accurate anycast simulation — ensuring the final implementation remained both reproducible and aligned with real-world Internet routing behavior .

Results and Observations

- **Key metrics and results**

Test	Metric	Observation	Note
BGP Session Establishment	Peering time	Established within 3 s	Stable connectivity between nodeA and nodeB verified via FRR logs
Flip Emulation	RTT penalty	+50 ms applied to nodeB	Emulated flippings using Mahimahi delay shells

- Bgp session establishment (network interaction of docker containers where routing using border-gateway-protocol)



- Flip emulation (using approach 1) :
  - Server with 30ms Delay

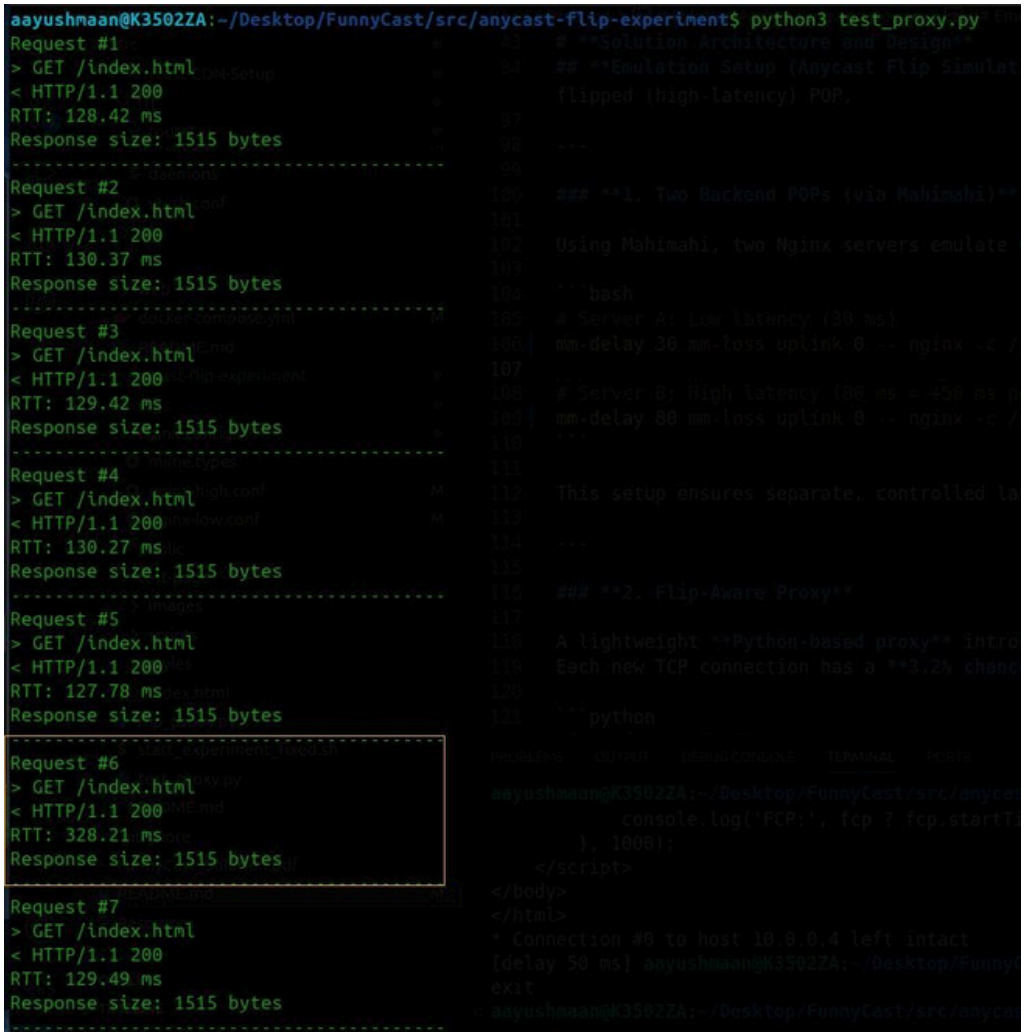
```
aayushmaan@K3502ZA:~/Desktop/FunnyCast/src/anycast-flip-experiment$ mm-delay 30
[delay 30 ms] aayushmaan@K3502ZA:~/Desktop/FunnyCast/src/anycast-flip-experiment$ nginx -c nginx-configs/nginx-low.conf -p .
[delay 30 ms] aayushmaan@K3502ZA:~/Desktop/FunnyCast/src/anycast-flip-experiment$ curl -v http://10.0.0.2:8081
*   Trying 10.0.0.2:8081...
* Connected to 10.0.0.2 (10.0.0.2) port 8081
> GET / HTTP/1.1
> Host: 10.0.0.2:8081
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.24.0 (Ubuntu)
< Date: Wed, 05 Nov 2025 13:30:54 GMT
< Content-Type: text/html
< Content-Length: 1515
< Last-Modified: Wed, 05 Nov 2025 13:17:07 GMT
< Connection: keep-alive
< ETag: "690b4e53-5eb"
< X-Backend-Type: low-latency
< X-Server-IP: 10.0.0.2
< Accept-Ranges: bytes
<
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Anycast Flip Test Page</title>
```

  - Server with 80ms Delay

```
aayushmaan@K3502ZA:~/Desktop/FunnyCast/src/anycast-flip-experiment$ mm-delay 80
[delay 80 ms] aayushmaan@K3502ZA:~/Desktop/FunnyCast/src/anycast-flip-experiment$ nginx -c nginx-configs/nginx-high.conf -p .
[delay 80 ms] aayushmaan@K3502ZA:~/Desktop/FunnyCast/src/anycast-flip-experiment$ curl -v http://10.0.0.4:8082
*   Trying 10.0.0.4:8082...
* Connected to 10.0.0.4 (10.0.0.4) port 8082
> GET / HTTP/1.1
> Host: 10.0.0.4:8082
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.24.0 (Ubuntu)
< Date: Wed, 05 Nov 2025 13:31:52 GMT
< Content-Type: text/html
< Content-Length: 1515
< Last-Modified: Wed, 05 Nov 2025 13:17:07 GMT
< Connection: keep-alive
< ETag: "690b4e53-5eb"
< X-Backend-Type: low-latency
< X-Server-IP: 10.0.0.4
< Accept-Ranges: bytes
<
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Anycast Flip Test Page</title>
```

  - Probabilistical flipping (~ 3.2%)

```
aayushmaan@K3502ZA:~/Desktop/FunnyCast/src/anycast-flip-experiment$ python3 flip_proxy.py
Flip proxy listening on 0.0.0.0:8080
 -> 96.8% to 10.0.0.2:8081
 -> 3.2% to 10.0.0.4:8082
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:30] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:31] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:31] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:31] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:31] "GET /index.html HTTP/1.1" 200 -
Flipped
127.0.0.1 - - [05/Nov/2025 19:02:31] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:31] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:32] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:32] "GET /index.html HTTP/1.1" 200 -
Not Flipped
127.0.0.1 - - [05/Nov/2025 19:02:32] "GET /index.html HTTP/1.1" 200 -
Not Flipped
```



- **Lessons Learned**
  - Studying the MAPRG problem statement and RFCs 4786 and 7094 highlighted the need to align our simulation with real anycast routing behavior.
  - Moving from simple latency emulation to BGP-driven flipping improved fidelity and taught us how routing policies influence performance.
  - Integrating Mahimahi, FRR, and Docker required detailed understanding of networking namespaces, routing tables, and container interconnects.
  - Implementing BGP sessions, route control, and latency injection deepened our practical knowledge of network configuration, diagnostics, and traffic engineering.
- **Future Work**
  - Expand analysis to include full Web Vitals (LCP, CLS, INP) for a deeper view of user experience under anycast flips.
  - Scale experiments with more vantage points and varied network types (4G, Wi-Fi, fiber).
  - Automate flip control to test different probabilities and regional failover patterns.

AIORI-2: Reporting and Standards Mapping

Team Name	Institution	Project Title	Focus Area
FunnyCast	Heritage Institute of Technology, Kolkata	Anycast Flipping & CDN User Experience	□ DNSSEC ☑ RPKI □ QUIC □ Encrypted DNS □ Other

Date: 05/11/2025

- **Standards Reference**

RFC / Draft No.	Title / Area	Lifecycle Stage	How This Work Relates
RFC 4786 / draft-ietf-grow-anycast	Operation of Anycast Services	⊘ Internet-Draft □ Proposed Standard □ Internet Standard	Implements or validates Section 4.3 of RFC

- **Impact on Standards Development**

Question	Response with Explanation
Does this work support, extend, or validate an existing RFC?	Yes. This work supports ongoing discussions in the IETF MAPRG related to routing stability and CDN performance but does not directly implement a specific RFC. It complements RFCs concerning Anycast deployment (RFC 4786) by empirically analyzing the performance implications of route instability on user experience.
Could it influence a new Internet-Draft or update sections of an RFC?	Potentially yes. The findings on flip-induced latency and FCP degradation can inform future Internet-Drafts addressing operational best practices for Anycast routing, CDN resilience, and transport-layer optimization under dynamic routing conditions.
Any feedback or data shared with IETF WG mailing lists (e.g., DNSOP, SIDROPS, DPRIVE, QUIC)?	Planned submission of results and dataset to the MAPRG mailing list for feedback.
Planned next step (e.g., share measurement dataset / open PR / draft text).	Publish dataset, share scripts, and draft a short Internet-Draft summarizing findings.

- **References**

- Agenda IETF122: maprg
  - https://datatracker.ietf.org/doc/agenda-122-maprg/
  - Mahimahi – http://mahimahi.mit.edu/
- Research References
  - V. Bajpai & J. Schönwälder, A Survey on Anycast in IPv6
  - Networks, IEEE Communications Surveys & Tutorials, 2017.
  - https://doi.org/10.1109/COMST.2017.2732261
- Cloudflare Blog
  - How Anycast Routing Improves Performance
  - and Resilience. https://blog.cloudflare.com/anycast
  - Akamai Research – Content Delivery and Web Performance
  - Insights. https://www.akamai.com/resources/research
- Additional Research References -
  - RFC 4786 – Operation of Anycast Services
  - IMC 2015 Paper – Measuring Anycast Performance
  - Core Research Paper – Anycast Analysis and Modeling
  - Anycast CDN Overview – Anycast.com
  - Web Vitals – Core Performance Metrics
  - HTTP/1.1 vs HTTP/2 – DigitalOcean Guide
  - Zhang et al., Duke University – Anycast Path Stability
  - High Performance Browser Networking – NGINX eBook

Acknowledgments

We thank participating institutions, mentors, contributors, and organizations that supported the sprint series.

Reflections from the Team

- **Aayushmaan (Team Lead):**
  - "Building this project made me appreciate how complex and dynamic Internet routing truly is. Simulating anycast flips at both the emulation and BGP levels helped me understand the subtle interplay between routing protocols and user-perceived web performance. Working with FRR and Docker taught me the value of reproducibility and controlled experimentation in network research."
- **Sanchayan Khan (Developer):**
  - "This project was a deep dive into real-world networking behavior. Integrating tools like Mahimahi, FRR, and connmark-based routing showed how even minor changes in path selection can significantly impact CDN performance. Debugging the container networks and emulating realistic anycast flips gave me a hands-on understanding of Internet-scale routing systems."

About the Authors

FunnyCast represents Heritage Institute of Technology, part of the AIORI-2 Hackathon (Nov 2025). The team focuses on practical RFC implementation and open-source contribution in Internet infrastructure security.

Contact

Lead Author: Aayushmaan **Email:** aayushmaan.cse27@heritageit.edu.in **Mentor:** Jhalak Dutta