Project report for Ripe Network Analysis

Christoph Hüttner BSc, Daniel Kornfeld, Marko Mrsic
Supervised by Postdoc-Ass. Priv.-Doz. Dr. Dragi Kimovski
and Univ.-Ass. Dipl.Ing. Kurt Klaus Horvath, BSc

Abstract

In this report, we introduce an approach for analysing popular internet service availability in 178 countries. Our implementation leverages the Ripe Atlas network, which consists of over 12,000 probes across all seven continents. We select the top 100 internet services from the research-oriented Tranco list and utilize the Ripe network to test their reachability in each country. Our evaluation shows that Fiji, along with Serbia and China, block or fail to reach the most services in our measurements. Furthermore, services from Google have the lowest availability in Asian countries, especially the main domain `google.com`. In contrast, Asian services across the board have high availability in western countries. The reliability of our measurements suffered under the usage of IPv4 and limited traceroute visibility, the general trends, however, could still be accurate.

1 Introduction

The global internet is often treated as a uniform communication platform, where popular services are assumed to be reachable from almost anywhere. In practice this assumption does not always hold. Reachability can vary significantly depending on the location from which a service is accessed. These differences may be caused by technical factors such as routing behavior, load balancing, or temporary outages [1]. In other cases they may be related to systematic restrictions or filtering that affect certain targets or protocols [2, 3]. Even if the exact reason for a failure cannot always be identified, large scale measurements can still reveal patterns that are unlikely to be caused by random effects alone [2].

The main research question of this work is whether the reachability of popular Internet services differs noticeably between countries when measured from many distributed vantage points. We are particularly interested in whether some countries show consistently higher numbers of unreachable services than others, and whether certain services exhibit strong regional differences. This question is relevant because popular services often provide essential infrastructure for communication, content delivery, and cloud based applications [4]. If such services are regularly unreachable from specific regions, this can affect users as well as systems that rely on globally available connectivity [1].

From a political perspective, Internet access has increasingly become a tool of governance and control. Governments may restrict access to foreign platforms to enforce domestic regulations, protect national industries or limit the spread of information that is perceived as politically sensitive [5]. Such restrictions are not a new phenomenon, as national firewalls and content filtering systems have existed for decades [2]. However, the scale, technical sophistication and normalization of these mechanisms have grown substantially [5]. Modern censorship techniques operate at multiple layers of the network stack and often remain opaque to end users [3]. This evolution raises concerns about the long term openness of the Internet and the risk of further fragmentation into regionally isolated networks [1].

The risk associated with this development extends beyond individual user experience. Reduced accessibility of global services impacts academic collaboration, economic participation and freedom of information [5]. From a technical standpoint, increasing heterogeneity in reachability complicates the design of globally reliable distributed systems [1]. This work contributes to the empirical understanding of these issues by providing a measurement based analysis of service availability using a globally distributed measurement infrastructure [6].

Several existing measurement platforms could in principle be used to study global reachability. For example, OONI provides a distributed infrastructure to detect network interference using specialized application-level tests [3]. Similarly, Censored Planet operates a longitudinal measurement platform that continuously monitors selected services from many locations [7]. While these systems provide detailed insight into specific interference mechanisms, they are mainly designed for targeted censorship detection and focus on predefined services or protocols. They are therefore less flexible for running broad and uniform reachability measurements across an arbitrary set of popular domains.

To study this question, we need a measurement infrastructure that offers geographically diverse vantage points and supports reproducible active probing of Internet services. Local test setups or measurements from only a single country are not sufficient to capture global reachability patterns, since they cover only a limited part of the possible network paths. RIPE Atlas overcomes this limitation by providing an openly accessible network of distributed probes deployed in many countries and networks [6]. This allows us to run the same measurements from many independent locations under comparable conditions.

Another reason for choosing RIPE Atlas is its standardized measurement environment. Basic measurement types such as ping and traceroute are executed in a consistent way across all probes, which reduces methodological differences and makes results easier to compare at large scale [6]. In addition, RIPE Atlas offers a public API that enables automated experiment control and data collection. This is essential for running measurement campaigns of the size required in this work. While RIPE Atlas is not specifically built to detect blocking or interference, its global coverage and ease of use make it a practical choice for comparative reachability studies.

Instead of testing full application-level functionality, we focus on a simple connectivity signal using ICMP-based ping measurements. This allows us to perform measurements at scale and apply the same method to many targets and locations. At the same time, we are aware that ping is a limited indicator. Some services block ICMP while remaining reachable via other protocols, and different probes may resolve the same domain to different IP addresses [3]. Despite these limitations, ICMP measurements provide a useful first overview and help identify services and regions that deserve closer analysis [2].

2 Related Work

Measuring Internet reachability and access limitations has been an active area of research for many years. Early studies often relied on controlled experiments or a limited number of region specific vantage points, which restricted their ability to capture global patterns. More recent work increasingly makes use of large scale measurement infrastructures and widely distributed probes in order to observe reachability differences across many countries and networks. This shift enables comparative analyses that focus on systematic effects rather than isolated incidents.

Le Pochat et al. [4] introduced the Tranco ranking as a research oriented alternative to traditional popularity lists. By aggregating multiple independent data sources, Tranco reduces susceptibility to manipulation and short term fluctuations. This makes it particularly suitable for longitudinal and comparative studies of Internet services. In the context of our work, the Tranco list provides a stable and reproducible set of targets that reflect real world usage patterns rather than transient trends, which is important for evaluating service reachability at scale.

The RIPE NCC staff [6] developed the Ripe Atlas network as a global platform for active Internet measurements. Atlas consists of thousands of distributed probes that can perform measurements such as ping, traceroute, and DNS queries toward user specified targets. The platform has been widely used to study latency, routing behavior, and general service reachability. While Ripe Atlas was not designed specifically for detecting blocking or censorship, its broad geographic coverage and standardized measurement interface make it well suited for comparative reachability studies across many countries.

Several studies have used large scale active measurements to investigate global reachability limitations and service blocking. Pearce et al. [2] conducted a large scale study on DNS manipulation and showed that reachability failures often correlate with geographic and regulatory boundaries rather than with purely random network failures. Their work emphasizes that different technical mechanisms can lead to similar observable failures, which complicates attribution and motivates conservative interpretation of measurement results.

Filastò and Appelbaum [3] presented an architecture for distributed interference measurement that highlights the difficulty of distinguishing between intentional blocking and incidental network problems. They argue that measurement results must be interpreted carefully and that simple reachability signals should be used primarily to identify anomalies rather than to infer exact mechanisms. Similar concerns apply to our work, particularly given the heterogeneous deployment environments of Ripe Atlas probes.

Agrawal et al. [8] focus on DNS based blocking techniques and show how different DNS failure modes correspond to distinct technical implementations of access restrictions. These findings underline the importance of name resolution in service reachability and provide useful context for interpreting failed measurements.

Raman et al. [7] present Censored Planet, a longitudinal measurement platform designed to continuously monitor service availability worldwide. Their work documents increasing deployment of interference techniques over time and demonstrates the value of sustained measurements for identifying long term trends. Unlike Ripe Atlas, Censored Planet is explicitly designed to detect and classify blocking behavior using multiple measurement signals.

Compared to these approaches, our work focuses on a comparatively simple but scalable reachability signal, namely ICMP based probing, applied uniformly across a large set of countries and services. While this approach does not capture the full complexity of modern interference techniques, it enables a broad comparative analysis using an existing and widely deployed measurement infrastructure. Rather than aiming for fine grained classification of blocking mechanisms, our work focuses on identifying systematic reachability differences and discussing them from a network perspective, including factors such as filtering behavior, DNS resolution effects, and infrastructure concentration.

3 Methodology

In this section, we describe the measurement methodology used in our study and explain how reachability results obtained from Ripe Atlas are interpreted. The goal of this methodology is to provide a clear and reproducible definition of service reachability that is suitable for large scale measurements while acknowledging the limitations of the underlying signals.

3.1 Measurement Setup

For target selection, we rely on the Tranco ranking, which aggregates multiple independent popularity sources and is commonly used in measurement research. From a snapshot taken on July 8th, 2025, we select the top 100 domains. These domains represent widely used Internet services across different regions and application categories, making them a suitable basis for studying global service availability. An excerpt of the selected targets is shown in Table 1, while the full list of domains is provided as part of our experiment artifacts.

As vantage points, we use RIPE Atlas probes that are reported as connected and that have a valid ISO two-letter country code assigned by the Atlas API. At the time of our measurement campaign, the RIPE Atlas network provided more than 12,000 connected probes distributed across 178 countries. From this pool, one probe is requested for each country, and for each measurement batch we request up to 25 probes. When more probes are available for a given country, RIPE Atlas randomly selects one probe. This approach provides broad geographic coverage while keeping the number of measurements per target manageable.

Each measurement consists of an ICMP ping from a probe to a target domain. Before sending the ping, Ripe Atlas resolves the domain name using the resolver configured on the probe. The resolved IPv4 address is then used as the destination for the ICMP echo requests. All measurements are restricted to IPv4 in order to ensure consistency across probes and targets and to avoid differences caused by uneven IPv6 deployment.

Rank	Domain
1	google.com
2	mail.ru
3	microsoft.com
4	facebook.com
5	dzen.ru
6	amazonaws.com
7	apple.com
8	googleapis.com
9	youtube.com
10	cloudflare.com

Table 1: Excerpt of the Tranco top sites used as measurement targets (snapshot July 8th, 2025).

3.2 DNS Resolution and Reachability

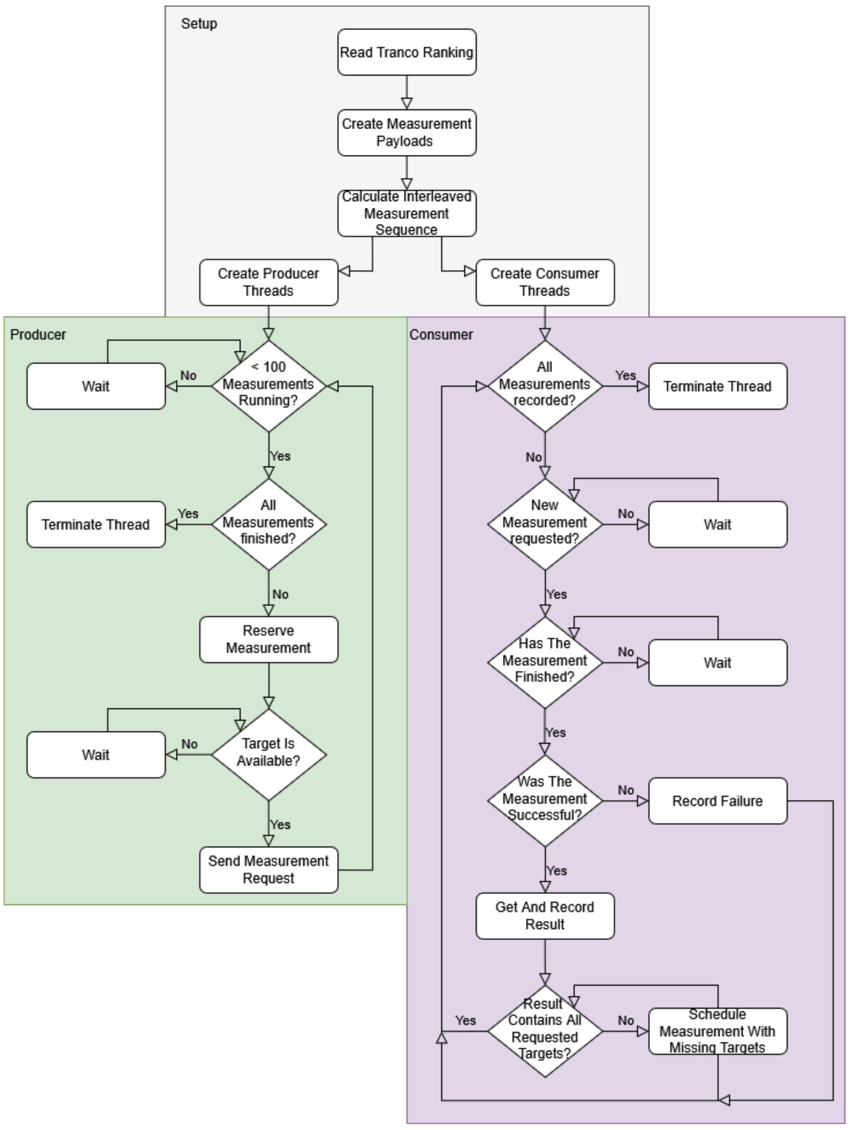
Our measurement approach implicitly relies on DNS resolution, since targets are specified as domain names rather than fixed IP addresses. From the perspective of our analysis, DNS resolution is treated as an internal step performed by Ripe Atlas. We observe only the resulting destination address and whether the probe receives any ICMP echo replies. As a result, our data does not explicitly encode whether a failure occurred during DNS resolution or during the network communication to the resolved address.

In our interpretation, a measurement is considered successful if at least one ICMP echo reply is received by the probe. A measurement is considered unsuccessful if no reply is received, regardless of whether this is due to DNS resolution failure, filtering of ICMP traffic, unreachable routing, or target side configuration. This design choice simplifies the analysis and aligns with the goal of capturing a coarse but scalable reachability signal.

It is important to emphasize that DNS reachability and network reachability are not equivalent. A domain may successfully resolve to an IP address while the subsequent ICMP traffic is filtered or dropped. Conversely, a failure during DNS resolution prevents any network level probing from taking place. Since Ripe Atlas does not expose detailed DNS error modes in a uniform way for all probes, we do not attempt to distinguish between these cases. Instead, we interpret unsuccessful measurements conservatively and avoid attributing them to a specific mechanism without additional evidence.

4 Approach

We leveraged insights from related work to create an asynchronous measurement program, which utilizes the consumer-producer pattern to create measurements and consume their results. A flowchart of this process can be seen in Figure 1.



Setup
Read Tranco Ranking
Create Measurement Payloads
Calculate Interleaved Measurement Sequence
Create Producer Threads
Create Consumer Threads
Producer
Wait
No
< 100 Measurements Running?
Yes
Terminate Thread
Yes
All Measurements finished?
No
Reserve Measurement
Wait
No
Target Is Available?
Yes
Send Measurement Request
Consumer
All Measurements recorded?
Yes
Terminate Thread
No
New Measurement requested?
No
Wait
Yes
Has The Measurement Finished?
No
Wait
Yes
Was The Measurement Successful?
No
Record Failure
Yes
Get And Record Result
Yes
Result Contains All Requested Targets?
No
Schedule Measurement With Missing Targets

Figure 1: Flowchart of the measurement program.

The flowchart is structured into the initial setup phase and the concurrent execution roles of producers and consumers. They coordinate the creation and collection of the measurements.

In the initial setup phase the program begins by reading the Tranco Ranking list, which is provided via a local file. This file is already preprocessed, to exclude domains that are known to be unsuitable for ICMP-based probing, such as services hosted exclusively on content delivery networks that do not respond to ping requests. Upon a successful read of the ranking list, the measurement payloads can be created. In addition, the interleaved measurement sequence is computed for efficient scheduling to overcome limitations of the RIPE Atlas API. Once these preparatory steps are completed, producer and consumer threads are initialized to execute the measurement campaign concurrently

The producer threads are responsible for submitting new measurement requests to RIPE Atlas. Firstly, the producer thread has to check whether there are less than 100 measurements currently running. If so, the flow can proceed. Otherwise, the producer has to wait to stay within the limitations of parallel measurements that RIPE Atlas has set. After proceeding, there needs to be another check to ensure that not all measurements have been completed. If they are finished, the thread can be terminated. When there are still measurements that still need to be made, a measurement is being reserved. However, before sending the request, it has to be made sure that the target is available. If not, the code waits. Otherwise, the request is sent.

The consumer threads handle the retrieval and storage of measurement results. Each consumer begins by repeatedly checking whether all measurements have already been recorded. If yes, the thread terminates. Otherwise, it waits for newly issued measurements to appear and periodically queries their execution status. Moreover, the consumer needs to know if a measurement was requested, whether it finished and also if it was successful. This prevents the thread to run into errors when consuming. If it failed the failure has to be recorded and we can reschedule a new measurement with all the missing targets to get as much data as possible. Thereby, we are not solely relying on a single measurement. However, if the measurement was successful the result is being received and recorded in the database. Finally, we have to make sure that the result contains all requested targets. If this condition is not met, a reschedule needs to happen. In contrast if everything is contained, the thread is finished and can terminate.

4.1 Funding

Using the Ripe Atlas network incurs a cost: each non-repeating ping costs 6 credits. With the goal of pinging 100 targets in each of 177 countries, the total cost amounts to 106,200 credits, as shown in Equation 1. Univ.-Ass. Dipl.Ing. Kurt Klaus Horvath provided us with 300,000 credits, and we additionally received 30 million credits from Clemens Bauer BSc. This funding allowed us to execute multiple measurement runs.

$$cost = countries \times targets \times costPerPing = 177 \times 100 \times 6 = 106{,}200\ Credits \tag{1}$$

4.2 Ripe Atlas Measurements

To access the Ripe Atlas network, we used their API[1] to create measurements. The service expects JSON-encoded target definitions, which we generate by reading URLs from the Tranco ranking, as seen in Listing 1. Using Tranco is particularly relevant in our context because it is constructed to be robust against manipulation, stable over time, and reproducible by citing a specific snapshot. This reduces the risk that our measurements focus on artificially promoted or short-lived domains.

On the vantage-point side, we query the Ripe Atlas API for all probes that are currently connected and have a valid country code (Listing 7). For each target, we then ask Atlas to select randomly one probe per country. This strategy gives us wide geographic coverage while staying within the credit limits of our campaign.

We then create a measurement payload consisting of one target and up to 25 probes per batch, as seen in Listing 2. This reduces the number of measurement requests from 17,700 to only 800, as shown in Equation 2.

[1] `https://atlas.ripe.net/docs/apis/rest-api-reference/`

```java
BufferedReader br = new BufferedReader(new FileReader(TARGET_URLS));

String line;

while ((line = br.readLine()) != null) {
    Map<String, Object> definition = new HashMap<>();
    definition.put("target", line);
    definition.put("type", "ping");
    definition.put("af", 4);
    definition.put("is_oneoff", true);

    targets.add(definition);
}
```

Listing 1: This code snippet initializes a buffered reader to process each entry of the Tranco target list sequentially. For every URL read from the input file, a JSON-like map structure is constructed to define the measurement parameters. The "target" field specifies the hostname to be pinged, "type" denotes the measurement method, and "af" enforces IPv4 connectivity. The "is_oneoff" flag ensures that each measurement runs only once, preventing redundant or looping requests. These definitions are collected in memory, preparing a structured set of tasks that can be efficiently distributed to probes later in the measurement pipeline.

```java
Map<String, Object> payload = new HashMap<>();
List<Map<String, Object>> probeList = new ArrayList<>();

for (int i = startCountryIndex; i < Math.min(countryCodesWithProbes.length,
↪   startCountryIndex + BATCH_SIZE); i++) {

    Map<String, Object> probe = new HashMap<>();
    probe.put("requested", 1);
    probe.put("type", "country");
    probe.put("value", countryCodesWithProbes[i]);

    probeList.add(probe);
}

payload.put("definitions", List.of(targets.get(targetIndex)));
payload.put("probes", probeList);

```

Listing 2: This section of the implementation generates the request payloads sent to the Ripe Atlas API. Each payload batches one target definition together with a list of probes selected by country. The batching strategy, which groups up to 25 probes per request, exploits a feature of the Ripe Atlas network. Bundling 25 probes to one target counts as one measurement, the reverse would count as 25. So batching 25 probes with one target drastically reduces the total number of measurements and HTTP requests, as described in Equation 2, without compromising the global coverage of measurements.

$$measurementRequests = targets \times \lceil \frac{countries}{batchSize} \rceil = 100 \times \lceil \frac{177}{25} \rceil = 800\ Requests \tag{2}$$

4.2.1 Measurement Scheduling

Ideally, the 800 requests could be sent simultaneously. However, our API token is limited to 100 concurrent measurement requests. Sequentially processing measurements would require waiting for each batch to finish before starting the next, which proved infeasible due to time constraints (Figure 2).

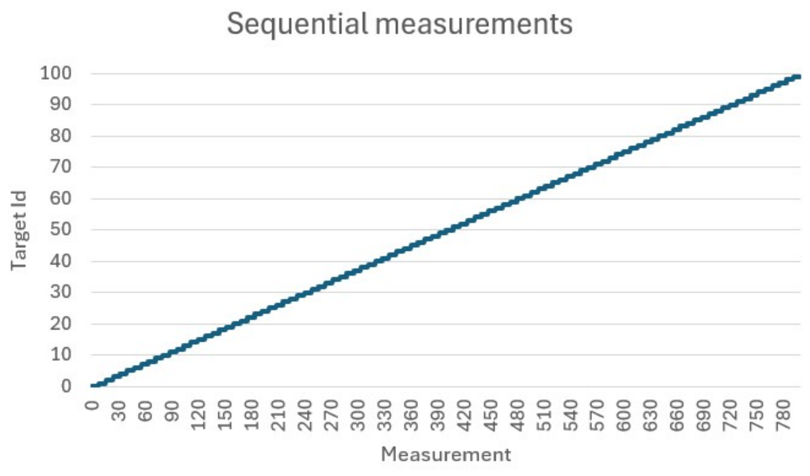


Figure 2: Sequential measurement requests.

To overcome this, we developed an interleaved scheduling algorithm (Figure 3), which sends multiple requests while respecting concurrency limits. Eight batches are required to measure one target across 177 countries, producing eight corresponding spikes in the graph. This reduces the number of waits from 800 to only 8. Wait times range between one and ten minutes, resulting in approximately one hour to measure 100 targets with 177 countries, compared to four days using the sequential approach.

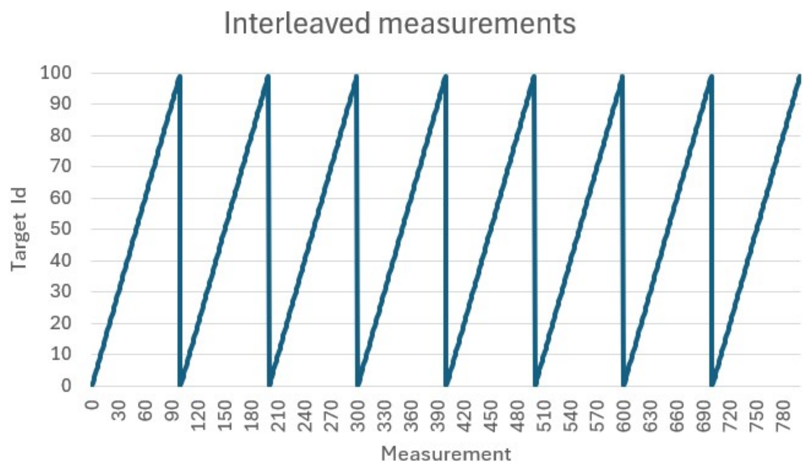


Figure 3: Interleaved measurement requests.

4.2.2 Producer-Consumer pattern

If 100 measurements are already running, our algorithm waits for them to finish and stores the result. This producer-consumer pattern necessitated the use of multiple threads, which we synchronize with atomic integers, concurrent hash maps, and concurrent linked lists. Our producer threads are limited by the amount of active measurements, which are limited by the concurrent measurement limit (100), as seen in Listing 3.

Since we do not want the producer threads to create duplicated measurements, they acquire their measurement index from an atomic integer, as seen in Listing 5.

The producers then send the measurement request to the Ripe Atlas network via HTTPS. If the measurement got successfully requested, we record its identifier so the consumer threads are able to periodically request the measurement status, as seen in Listing 6.

Furthermore, the producers have to wait for the probe to finish its current batched measurement, so the number of measurements towards a given target do not exceed the limit (25), as seen in Listing 4.

```java
// Inside the producer thread
while (true) {
    int current = activeMeasurements.get();

    if (current >= CONCURRENT_MEASUREMENT_LIMIT) {
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
            throw new RuntimeException(e);
        }
        continue;
    }

    if (activeMeasurements.compareAndSet(current, current + 1)) {
        break;
    }
}
```

Listing 3: The producer thread continuously monitors the number of active measurements by polling an atomic integer counter. Once the active measurement count reaches the defined concurrency threshold, the thread temporarily sleeps before rechecking the limit. This approach ensures that the system respects the 100 concurrent measurement restriction enforced by the Ripe Atlas API, thereby avoiding rate-limit errors. The atomic compare-and-set operation guarantees thread-safe increments and prevents race conditions, ensuring stable multi-threaded performance under high measurement load.

```java
// Inside the producer thread
while (true) {
    if (targetIsInMeasurement.get()) {
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
            throw new RuntimeException(e);
        }
        continue;
    }
    if (targetIsInMeasurement.compareAndSet(false, true)) {
        break;
    }
}
```

Listing 4: In addition to the global concurrency cap, each target is also guarded by an atomic flag preventing multiple concurrent batches against the same host. This ensures that no more than 25 probes measure a single target simultaneously. The use of thread sleeping and atomic synchronization ensures fairness between threads and prevents any target from becoming a bottleneck or causing duplicate measurements. This design effectively balances workload distribution across thousands of measurements.

```java
int measurementIndex = traversalIndex.getAndUpdate((t) -> {
    if (t < traversalOrder.size()) {
        return t + 1;
    }
    return t;
});
```

Listing 5: Each producer thread acquires a unique measurement index from an atomic counter, ensuring that no two threads attempt to handle the same measurement. This guarantees deterministic traversal of the measurement order and facilitates reproducibility during debugging or repeated runs. Using an atomic integer for index progression eliminates the need for synchronized collections, thereby improving throughput while maintaining consistency.

```java
public static String getMeasurementStatus(CloseableHttpClient client, long
    ↪ measurementId) {
    String statusUrl = "https://atlas.ripe.net/api/v2/measurements/" +
        ↪ measurementId + "/";

    HttpGet statusGet = new HttpGet(statusUrl);
    statusGet.setHeader("Authorization", "Key " + apiKey);

    return client.execute(statusGet, (ClassicHttpResponse response) -> {
        int code = response.getCode();
        if (code != 200) {
            Log.error("Failed to fetch measurement status: HTTP " + code);
        }

        String body = EntityUtils.toString(response.getEntity());
        ObjectMapper mapper = new ObjectMapper();
        JsonNode root = mapper.readTree(body);
        JsonNode statusNode = root.path("status").path("name");

        if (statusNode.isMissingNode()) {
            Log.error("Measurement status not found in response.");
        }

        return statusNode.asText();
    });
}
```

Listing 6: The "getMeasurementStatus" method encapsulates the logic for querying the Ripe Atlas API about the current state of a specific measurement. It performs an authenticated GET request using the API key and parses the JSON response to extract the status name. The function handles potential response errors and logs unexpected conditions, improving fault tolerance.

4.2.3 Measurement Result

If the status text of the measurement is "Stopped", we get the result of the measurement and store it in a local SQLite database. Additionally, the consumer thread decrements the active measurement atomic integer, which enables the producer threads to send further requests. If the amount of batched measurement results are not equal to the requested amount, we reschedule the missing measurements.

Since the Ripe Atlas network contains probes with different firmware versions, the measurement results we receive range from firmware version 4790 to 5110, most have version 5080, as seen in Figure 4. We then store all fields defined by the Ripe Atlas result format[2] in our database.

We then get the country ISO2 code of the probes country, as seen in Listing 7. Additionally, we then added the target URL by calling another endpoint of the Ripe Atlas API, as seen in Listing 8.

[2]https://atlas.ripe.net/docs/apis/measurement-result-format/

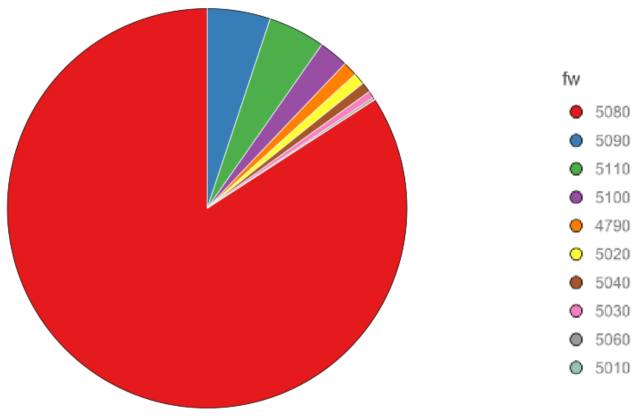


Figure 4: Distribution of firmware versions (fw) of measurement responses.

```java
public static Set<String> getConnectedCountryCodes() throws Exception {
    Set<String> countryCodes = new HashSet<>();
    ObjectMapper mapper = new ObjectMapper();
    String url = "https://atlas.ripe.net/api/v2/probes/?status_name=Connected
    &page_size=500";

    try (CloseableHttpClient client = HttpClients.createDefault()) {
        while (url != null) {
            HttpGet get = new HttpGet(url);
            String response = client.execute(get, (ClassicHttpResponse
            ↪  httpResp) ->
                    EntityUtils.toString(httpResp.getEntity())
            );

            JsonNode root = mapper.readTree(response);
            for (JsonNode probe : root.path("results")) {
                String cc = probe.path("country_code").asText();
                if (!cc.isEmpty()) {
                    countryCodes.add(cc);
                }
            }

            JsonNode next = root.path("next");
            url = next.isNull() ? null : next.asText();
        }
    }

    return countryCodes;
}
```

Listing 7: This method dynamically retrieves the list of all currently connected probes and their corresponding ISO country codes. By iterating through paginated API responses, it ensures complete coverage of all available probes without manual enumeration. The use of the Jackson JSON parser streamlines the conversion from raw API responses into structured Java objects. This functionality enables geographically distributed measurement scheduling, as each probe's country code becomes a routing key for assigning measurement tasks.

```java
public static void getRealTargetUrl() {
    ObjectMapper mapper = new ObjectMapper();
    HashMap<Integer, String> targetAddresses =
     ↪  SQLiteConnector.getTargetAddresses();
    HashMap<String, String> urlMap = new HashMap<>();
    for(int id : targetAddresses.keySet()) {
        String resultsUrl = "https://atlas.ripe.net/api/v2/measurements/" + id
         ↪  + "/";
        String dst_addr = targetAddresses.get(id);

        CloseableHttpClient client = HttpClients.createDefault();
        HttpGet get = new HttpGet(resultsUrl);
        get.setHeader("Authorization", "Key " + apiKey);

        TextNode result = null;
        try {
            result = client.execute(get, (ClassicHttpResponse response) -> {
                String body = EntityUtils.toString(response.getEntity());
                JsonNode root = mapper.readTree(body);
                return (TextNode)root.get("target");
            });
        } catch (Exception e) {
            e.printStackTrace();
        }

        urlMap.put(dst_addr, result.asText().replaceAll("\"", ""));
    }

    SQLiteConnector.attachUrlToDB(urlMap);
}
```

Listing 8: The "getRealTargetUrl" function assigns measurement identifiers with their actual destination URLs by querying the Ripe Atlas API for metadata about each measurement. It enhances the local database with mappings between measurement results and target hosts. This step is essential for post-analysis, as it enables aggregation and filtering of results based on target domains.

4.3 Initial Analysis

We copied the Tranco list on July 8th, 2025, and selected the top 100 services. Notable examples can be viewed in Table 1

After executing our measurement algorithm with 177 countries and the top 100 internet services provided by the Tranco list, we analysed the measurement results by grouping the failed measurements of each country, as seen in Listing 9.

This gives us a list of countries, along with the number of their failed requests. The top 10 countries can be seen in Figure 5. Fiji had the highest amount of failed requests with 58, followed by China with 25, and Serbia with 20. We therefore selected these three countries for further analysis.

Furthermore, we analysed the services with the lowest reachability, as seen in Figure 6. An important caveat to consider is our Ripe Atlas measurements only use IPv4, so these targets might still be reachable with IPv6. This might explain why `lencr.org`, `msedge.net`, `netflix.com`, and `t.me` are not reachable. Other services, like `microsoft.com` might block pings, which would further decrease their ranking in our analysis.

```sql
SELECT prb_country, COUNT(*) AS failed_count
FROM measurement_results
WHERE rcvd = 0 AND prb_country IS NOT NULL
GROUP BY prb_country
ORDER BY failed_count DESC;
```

Listing 9: The SQL query aggregates the number of failed ping responses per country, serving as a high-level indicator of national-level accessibility. By filtering for measurements where no response was received (rcvd $= 0$), the query highlights potential cases of network blocking, filtering, or service unavailability. Grouping and ordering the results by failure count makes it straightforward to identify regions with systematic connectivity issues. This analytical step represents the transition from raw measurement data to meaningful insight about internet openness.

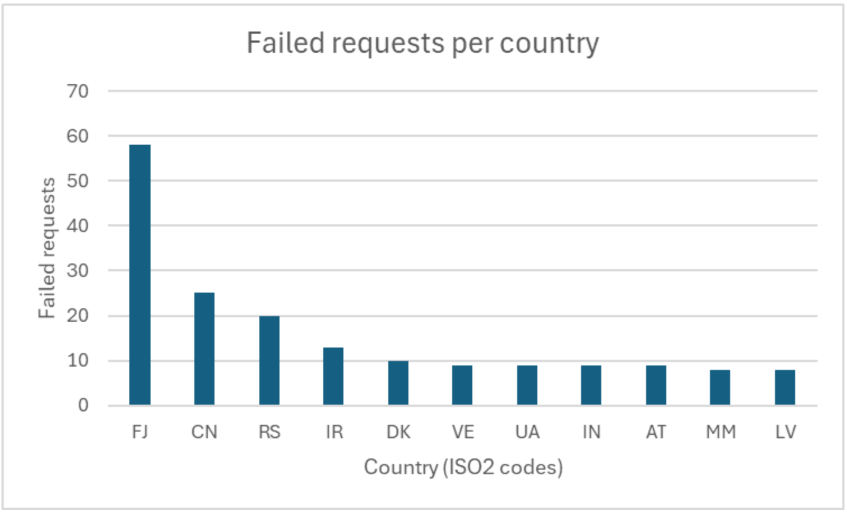
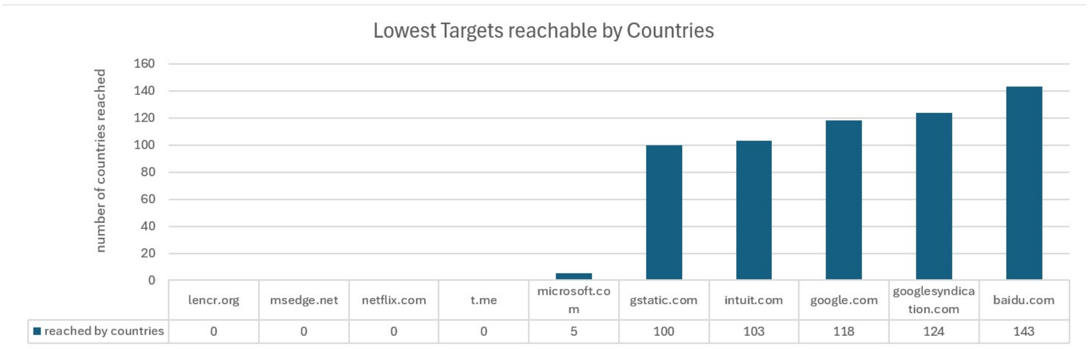


Figure 5: Failed requests per country in descending order, limited to the top 10.

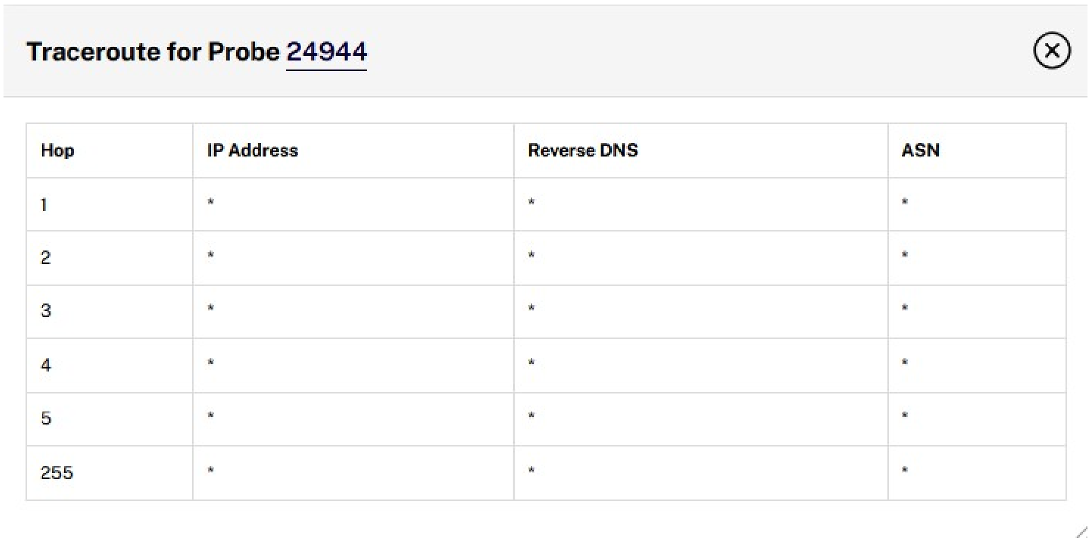


	lencr.org	msedge.net	netflix.com	t.me	microsoft.co m	gstatic.com	intuit.com	google.com	googlesyndica tion.com	baidu.com
■ reached by countries	0	0	0	0	5	100	103	118	124	143

Figure 6: Top 10 targets with the lowest reachability (IPv4 Pings).

4.4 Analysis of countries with low service availability

To figure out why Fiji has such a high fail rate, we sent traceroute measurements via the Ripe Atlas website that target those failed targets from a Fiji probe. Surprisingly, none of those measurements were able to get any information, except the amount of hops and that they failed (see Figure 7). We therefore think that there might be a filter on the Fiji probes or some form of configuration that prohibits such measurements.



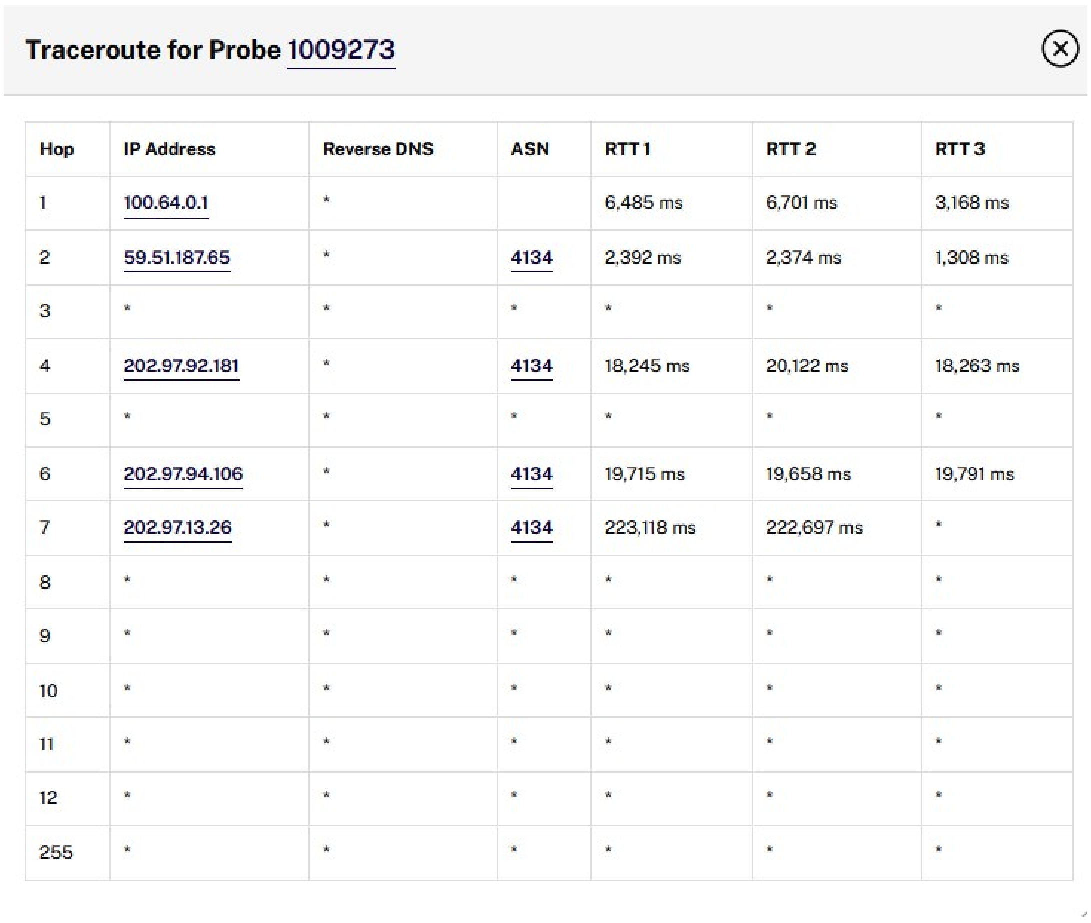
Traceroute for Probe 24944

Hop	IP Address	Reverse DNS	ASN
1	*	*	*
2	*	*	*
3	*	*	*
4	*	*	*
5	*	*	*
255	*	*	*

Figure 7: Traceroute result of whatsapp.com in Fiji.

In contrast, we were able to traceroute with Serbia probes, where we could not discover a single institution where traffic gets funnelled through and filtered, instead the requests ended at either the internet service providers, or certain backbone providers, such as Arelion.

Tracerouting with probes in China revealed that the requests end at the backbone of China's internet service providers, as seen in Figure 8. The request is not finished there, but additional hops do no reveal their address, likely due to filtering or lach of visibility. Other Google services, as well as WhatsApp, Twitter (X), and Facebook show similar results when measuring them in China.



Traceroute for Probe 1009273

Hop	IP Address	Reverse DNS	ASN	RTT 1	RTT 2	RTT 3
1	100.64.0.1	*		6,485 ms	6,701 ms	3,168 ms
2	59.51.187.65	*	4134	2,392 ms	2,374 ms	1,308 ms
3	*	*	*	*	*	*
4	202.97.92.181	*	4134	18,245 ms	20,122 ms	18,263 ms
5	*	*	*	*	*	*
6	202.97.94.106	*	4134	19,715 ms	19,658 ms	19,791 ms
7	202.97.13.26	*	4134	223,118 ms	222,697 ms	*
8	*	*	*	*	*	*
9	*	*	*	*	*	*
10	*	*	*	*	*	*
11	*	*	*	*	*	*
12	*	*	*	*	*	*
255	*	*	*	*	*	*

Figure 8: Traceroute result of google.com in China.

4.5 Analysis of popular western services in asian countries

To analyse the availability of popular Asian services in western countries, we filter the measurements in two ways. The targets get limited to Asian services and the countries of the probes get limited to western countries, as seen in Listing 10. baidu.com was reached by the least amount of countries (22), followed by dzen.ru, qq.com, tiktokv.com, and userapi.com, as seen in Figure 9. Overall, the services are reachable by nearly all western countries.

```sql
SELECT dst_name,
       SUM(CASE WHEN rcvd > 0 THEN 1 ELSE 0 END) AS success_count
FROM measurement_results
WHERE (
    dst_name = "mail.ru"
        OR dst_name = "dzen.ru"
        OR dst_name = "yandex.net"
        OR dst_name = "okcdn.ru"
        OR dst_name = "userapi.com"
        OR dst_name = "baidu.com"
        OR dst_name = "qq.com"
        OR dst_name = "tiktok.com"
        OR dst_name = "tiktokv.com"
        OR dst_name = "xiamo.com"
    )
  AND (
    prb_country = "AD" OR prb_country = "AT" OR prb_country = "BE" OR
    ↪  prb_country = "CH"
        OR prb_country = "DE" OR prb_country = "DK" OR prb_country = "ES" OR
        ↪  prb_country = "FI"
        OR prb_country = "FR" OR prb_country = "GB" OR prb_country = "IE" OR
        ↪  prb_country = "IS"
        OR prb_country = "IT" OR prb_country = "LI" OR prb_country = "LU" OR
        ↪  prb_country = "MC"
        OR prb_country = "MT" OR prb_country = "NL" OR prb_country = "NO" OR
        ↪  prb_country = "PT"
        OR prb_country = "SE" OR prb_country = "SM" OR prb_country = "VA" OR
        ↪  prb_country = "CA"
        OR prb_country = "US" OR prb_country = "AU" OR prb_country = "NZ"
    )
GROUP BY dst_name;
```

Listing 10: The SQL query aggregates the number of failed ping responses for Asian targets, limited to probes from western countries.

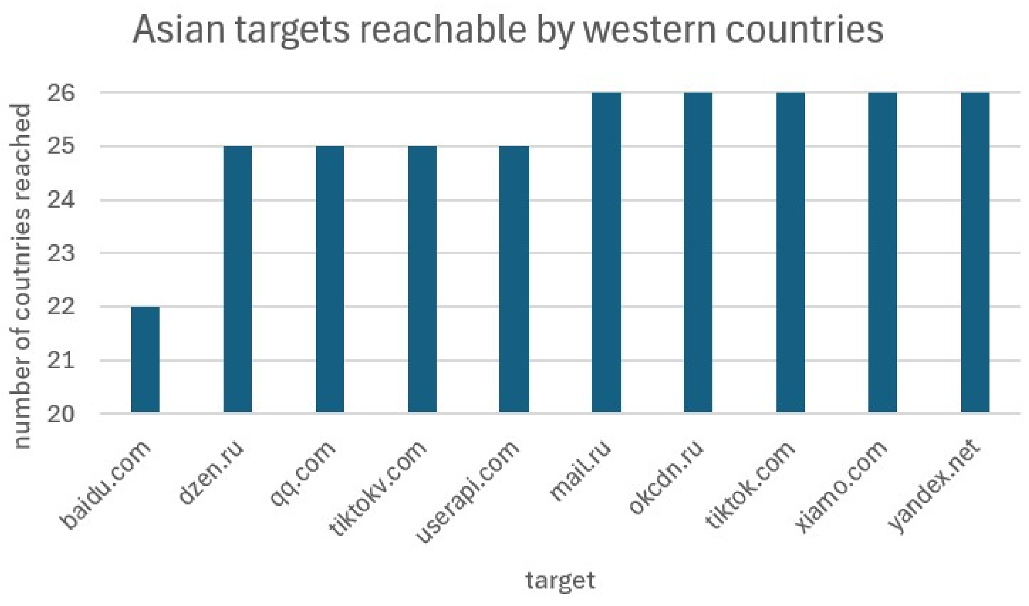


Figure 9: Reachability of Asian targets by western countries.

4.6 Analysis of popular Asian services in western countries

To analyse the availability of popular western services in Asian countries, we filter the measurements in two ways, just like in the previous section (Analysis of popular western services in asian countries) but in reverse. The targets get limited to Asian services and the countries of the probes get limited to western countries, as seen in Listing 11. Google services suffered the most, with google.com only reachable by 20 Asian countries, as seen in Figure 10.

```sql
SELECT dst_name,
       SUM(CASE WHEN rcvd > 0 THEN 1 ELSE 0 END) AS success_count
FROM measurement_results
WHERE dst_name IN (
                   "google.com", "facebook.com", "amazonaws.com", "apple.com",
                   "googleapis.com", "youtube.com", "cloudflare.com",
                   ↪  "instagram.com", "twitter.com",
                   "gstatic.com", "office.com", "azure.com", "linkedin.com",
                   ↪  "live.com",
                   "googlevideo.com", "googletagmanager.com", "fbcdn.net",
                   ↪  "amazon.com",
                   "wikipedia.org", "workers.dev", "doubleclick.net",
                   ↪  "github.com",
                   "googleusercontent.com", "whatsapp.net", "fastly.net",
                   ↪  "bing.com",
                   "wordpress.org", "sharepoint.com", "icloud.com",
                   "windows.net", "youtu.be", "skype.com", "pinterest.com",
                   ↪  "digicert.com",
                   "googlesyndication.com", "whatsapp.com", "roblox.com",
                   ↪  "goo.gl",
                   "adobe.com", "msn.com", "office365.com", "wordpress.com",
                   ↪  "mozilla.org",
                   "zoom.us", "cloudflare.net", "google-analytics.com",
                   ↪  "gandi.net",
                   "googleadservices.com", "googledomains.com", "opera.com",
                   "app-analytics-services.com", "windows.com", "snapchat.com",
                   ↪  "blogspot.com",
                   "unity3d.com", "cloudflare-dns.com", "reddit.com",
                   ↪  "intuit.com",
                   "nginx.org", "ui.com", "outlook.com", "sentry.io", "wa.me",
                   "nginx.com", "app-measurement.com", "dropbox.com",
                   ↪  "nist.gov",
                   "adnxs.com", "gravatar.com", "github.io", "apache.org",
                   "criteo.com", "applovin.com", "tumblr.com", "dns.google",
                   ↪  "f5.com"
    )
  AND prb_country IN (
                      "CN","JP","KR","TW","HK","MO",
                      "SG","MY","TH","VN","PH","ID","KH","LA","MM","BN","TL",
                      "IN","PK","BD","LK","NP","MV","BT",
                      "KZ","UZ","TJ","TM","KG",
                      "AE","SA","IL","IR","IQ","JO","LB",
                      "OM","QA","KW","YE","SY","TR"
    )
GROUP BY dst_name;
```

Listing 11: The SQL query aggregates the number of failed ping responses for western targets, limited to probes from Asian countries.

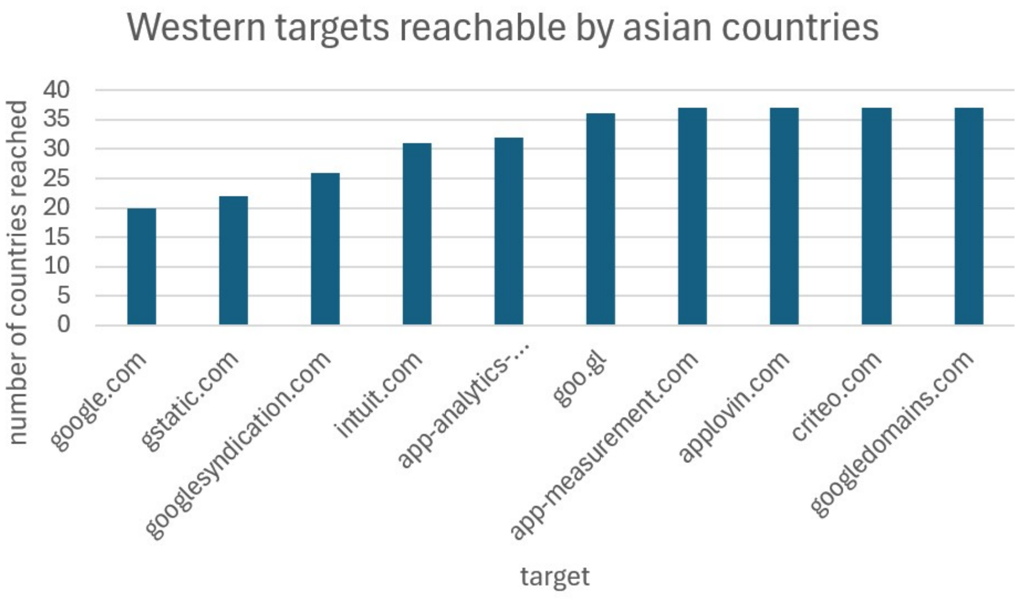


Figure 10: Reachability of western targets by Asian countries.

4.7 Accuracy of Ripe Atlas probes

Not all entries of the Tranco list are reachable using ICMP ping. For this reason, we filter out targets that are unreachable from all countries and restrict parts of our analysis to domains that are reachable from at least some locations. As described in the methodology, Ripe Atlas resolves domain names before issuing ICMP probes, which introduces several limitations that affect the interpretation of our results [6].

One important limitation arises from the distinction between DNS reachability and network reachability. We observe that DNS resolution often returns different IP addresses for the same domain depending on the geographic location of the probe, for example due to content delivery networks. Some of these resolved addresses respond to ICMP probes, while others do not. As a result, a domain may successfully resolve via DNS but still appear unreachable in our ICMP based analysis, even in the absence of deliberate blocking or filtering. This phenomenon has also been observed in prior large scale DNS measurement studies and complicates attribution of failures to specific causes [2].

A second source of ambiguity is the treatment of ICMP traffic by services and network operators. Many large platforms and CDNs intentionally drop or rate limit ICMP echo requests for operational or security reasons. In such cases, our measurements cannot distinguish between a service that is intentionally blocking access and one that simply does not respond to ping. Consequently, ICMP based reachability should be interpreted as a lower bound on actual service availability rather than as definitive evidence of blocking [3].

Path level measurements using traceroute could, in principle, provide additional insight into where packets are dropped. In practice, however, traceroute results are frequently incomplete or heavily censored. For example, in China we observe that traceroutes often terminate at national backbone providers, with subsequent hops not revealing any address information. Similar behaviour is observed for other destinations where filtering is suspected. While this suggests the presence of centralised traffic management, the exact mechanisms remain opaque and cannot be reliably inferred from traceroute data alone [7].

The suitability of Ripe Atlas probes also varies across regions. In some countries, such as Fiji, even benign traceroute measurements fail completely. This may be caused by probe sandboxing, restrictive ISP policies, or filtering of measurement traffic itself. In such environments, Ripe

Atlas provides limited visibility into path level behaviour and is therefore not an ideal vantage point for detailed diagnosis of reachability failures.

Finally, our measurement campaign is restricted to IPv4 connectivity. Several modern services increasingly deploy IPv6 first or IPv6 only endpoints in certain regions. As a result, services that are reachable over IPv6 may incorrectly appear unreachable in our dataset. This limitation may lead to an overestimation of unreachability for services that prioritise IPv6 deployment.

Taken together, these factors indicate that Ripe Atlas is not well suited for fine grained analysis of the precise causes of unreachability. Nevertheless, the platform remains highly valuable due to its global coverage, consistent measurement interface, and ease of deployment. While individual measurements can be ambiguous, aggregated results across many probes, targets, and countries still provide meaningful insight into large scale and systematic differences in service reachability.

5 Conclusion

In this work, we investigated the reachability of popular Internet services from a large number of geographically distributed vantage points using the Ripe Atlas measurement platform. By combining a reproducible target selection based on the Tranco ranking with large scale ICMP based measurements, we obtained a broad view of how service reachability differs across countries and regions. Our measurements demonstrate clear regional differences in accessibility. In particular, several globally dominant western services show consistently reduced reachability from parts of Asia, most notably from China, whereas popular Asian services remain reachable from nearly all western countries in our dataset.

This asymmetric reachability pattern suggests a technically fragmented global Internet in which access to services depends strongly on the vantage point. Networks with centralised backbone infrastructures or restrictive traffic handling policies tend to exhibit higher rates of unreachability, while networks in Europe, North America, and Oceania generally provide broad connectivity to foreign services. From a technical perspective, these differences are consistent with variations in routing behavior, filtering practices, and service side configuration choices rather than with random network failures.

When placing our findings in the context of related measurement studies, our results align with previous observations that documented systematic regional differences in service reachability. Large scale longitudinal measurement platforms have shown that such differences are often persistent and observable over long time spans [7]. DNS focused studies further demonstrate that reachability failures frequently depend on location and resolution behavior rather than on transient network outages [2]. While our study does not allow for a direct comparison with measurements from a decade ago, the fact that similar patterns remain observable today suggests that global Internet reachability has not converged toward a uniformly accessible environment.

At the same time, our analysis highlights important limitations. As discussed in the accuracy analysis, ICMP based probing provides only a coarse reachability signal and cannot distinguish between different underlying causes of unreachability. DNS resolution behavior, protocol specific filtering, and service side policies can all influence the observed results. For this reason, our findings should be interpreted as indicators of relative reachability differences rather than as definitive evidence of specific blocking mechanisms.

Several directions for future work follow naturally from these observations. Extending the measurement campaign to include IPv6 would reduce bias for services that prioritise modern protocol deployments. Incorporating explicit DNS measurements would allow separation of DNS related failures from network level effects. Application level probing, such as TCP connection attempts or TLS handshakes, could further improve the accuracy of reachability classification at the cost of increased measurement complexity. Repeating the measurements over longer time periods would enable longitudinal analysis and help distinguish persistent structural differences from temporary routing anomalies or outages. Finally, combining Ripe Atlas measurements with data from more specialised measurement platforms could provide complementary perspectives and strengthen confidence in the interpretation of observed patterns.

In conclusion, this work demonstrates that Ripe Atlas can be effectively used to study large scale service reachability when its limitations are carefully considered. Although the platform is not designed for fine grained diagnosis of interference mechanisms, its global coverage and ease of use make it a valuable tool for identifying systematic differences in Internet service accessibility across regions.

References

[1] A. Dainotti, C. Squarcella, E. Aben, *et al.*, "Analysis of country-wide internet outages caused by censorship," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 1–14, 2015.

[2] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Weaver, and V. Paxson, "Global measurement of dns manipulation," in *26th USENIX Security Symposium*, pp. 307–323, 2017.

[3] A. Filastò and J. Appelbaum, "Ooni: Open observatory of network interference," *USENIX ;login:*, 2012.

[4] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proceedings 2019 Network and Distributed System Security Symposium*, NDSS 2019, Internet Society, 2019.

[5] A. Master and C. Garman, "A worldwide view of nation-state internet censorship," *Proceedings on Privacy Enhancing Technologies*, 2023.

[6] R. N. Staff, "Ripe atlas: A global internet measurement network," *Internet Protocol Journal*, vol. 18, no. 3, pp. 2–26, 2015.

[7] R. S. Raman, N. Shenoy, K. Kohls, and R. Ensafi, "Censored planet: An internet-wide, longitudinal censorship observatory," *Proceedings of the ACM SIGCOMM Conference*, 2020.

[8] S. Agrawal, P. Verma, A. M. Kakhki, Z. Wang, and A. Dainotti, "Characterizing and measuring blocking of domains using dns," in *Proceedings of the ACM Internet Measurement Conference*, 2021.