



**RIPE
NCC**

Workshop: Advanced RIPE Atlas Usage

Daniel Quinn, Massimo Candela,
Chris Amin, Johan ter Beest, Philip Homburg
RIPE NCC

- Finding public measurements (5 minutes)
 - **Exercise: Analyse results** (10 minutes)

- Creating a measurement (5 minutes)
 - **Exercises: Create a measurement** (10 minutes)

- Integration with network monitoring systems (5 minutes)
 - **Exercise: Create status checks** (10 minutes)

- Real-time performance monitoring (5 minutes)
 - **Exercise: Use streaming API** (10 minutes)

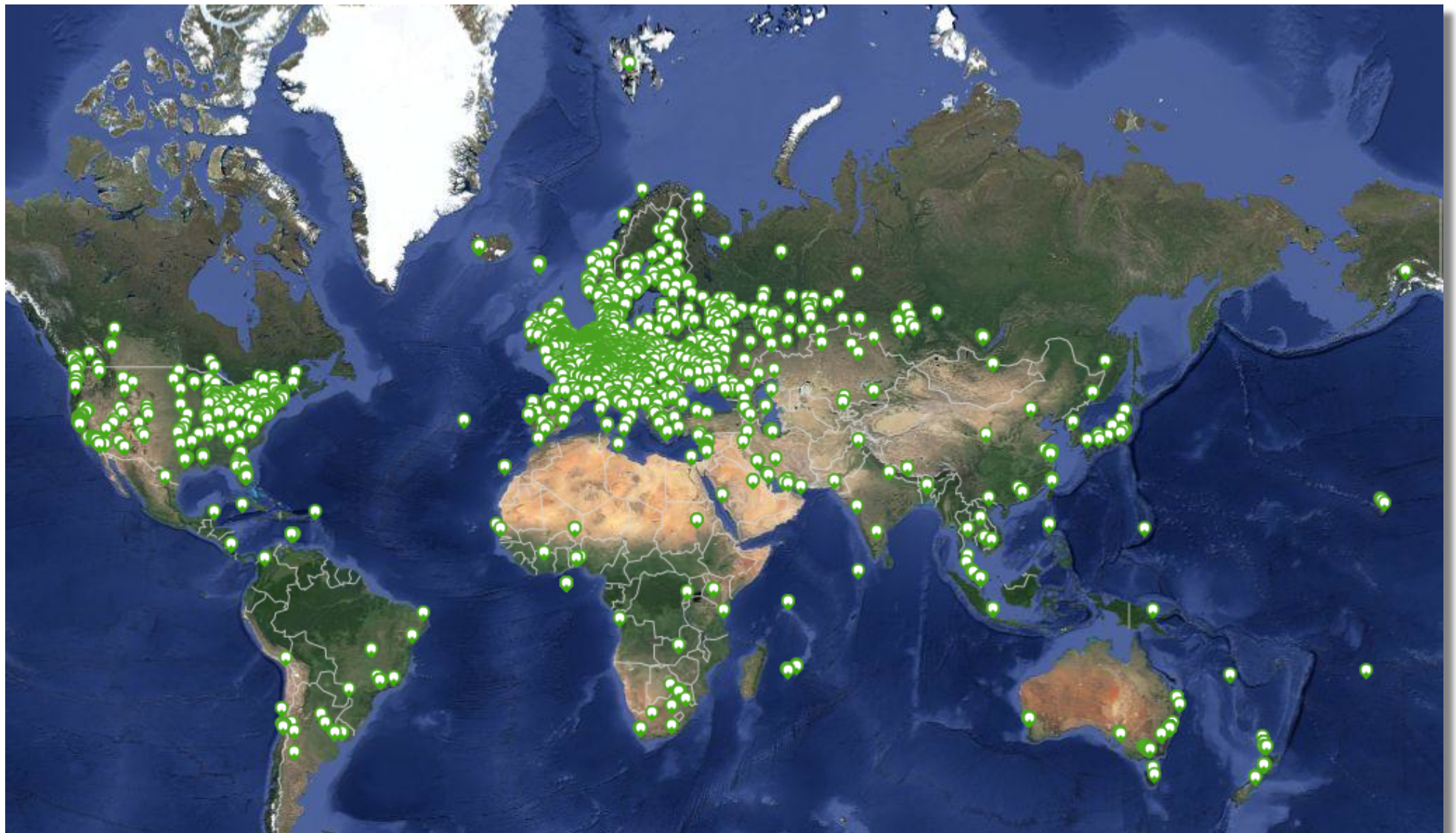
- How many of you have ever heard of RIPE Atlas?
- Who's used the graphical interface to schedule a measurement in RIPE Atlas?
- How many of you have used the RIPE Atlas API?
- Workshop description:
 - <https://ripe70.ripe.net/programme/meeting-plan/workshops/>



Introduction to RIPE Atlas



- RIPE Atlas = global active measurements platform
- Goal: View Internet reachability
- Probes hosted by volunteers
- Ongoing global measurements towards root name servers
 - Visualised as Internet traffic maps
- Ongoing regional measurements towards “anchors”
- Users can run customised measurements
 - Ping, traceroute, DNS, SSL/TLS and **NTP** (new)
- Data publicly available





Finding Results of Public Measurements



- There are many measurements already running!
- Search for existing public measurements first
- Schedule your own measurement if you don't find what you're looking for

- Log in to atlas.ripe.net with your RIPE NCC Access account
 - This is the same account used to access the LIR Portal, RIPE Atlas, RIPEstat, RIPE Labs...)
 - Create an account if you don't have one already

- Go to “My Atlas” > “Measurements”

The screenshot shows the RIPE Atlas website interface. At the top, there's a navigation bar with the RIPE NCC logo and a search bar. Below that, a breadcrumb trail reads: "You are here: Home > Analyse > Internet Measurements > RIPE Atlas > Measurements". A sidebar on the left contains a "My Atlas" dropdown menu with options like Probes, Measurements, Credits, API Keys, and Messages (72 new). The main content area is titled "Measurements" and features a filter bar with options for "Filter by target and/or description", "Any Status", "IPv4/v6", "All types" (which is expanded to show "Ping", "Traceroute", "DNS", "HTTP", and "SSL Certificate"), and "of all time". Below the filter bar, there are tabs for "Mine", "Favourites", "Hidden", "Public", and "All". A table of measurement results is displayed with columns for "Id", "Type", "Target", "Description", "(UTC)", and "Status". The first row shows a measurement with Id "1965015", Type "IPv4 ping", Target "b92.net", and Description "Ping measurement to b92.net".

Id	Type	Target	Description	(UTC)	Status
1965015	IPv4 ping	b92.net	Ping measurement to b92.net	2015-04-21 08:20 2015-04-21 08:30	■ 👁 ★

- Click on measurement, then “Download”

- Example:

- Target: www.seil.jp
- ID #1733329
- <https://atlas.ripe.net/measurements/1733329/>

- Results in JSON

You are here: Home > Data & Tools > RIPE Atlas > Measurements > Measurement #1733329

www.seil.jp

General Information Probes Map OpenIPMap Prototype **Download Results**

Download the raw measurement result data here.

You can use this form to download the data through your browser, or use the preview on the right to help you query the REST API directly.

Start Date:

Stop Date:

Format:

Download

URL Preview

```
https://atlas.ripe.net/api/v1/measurement/1733329/result/?start=1408924800&stop=1409011198&format=json
```

www.seil.jp

General Information **Probes** Map OpenIPMap Prototype Download Results

Probe	ASN (v4)	ASN (v6)		Time	RTT	Hops
2043	3313			2014-08-25 07:44	308.018	21
3246	41135			2014-08-25 07:41	259.912	12
3389	3302			2014-08-25 07:43	285.608	17
4092	37497			2014-08-25 07:40	452.889	19
4228	3269			2014-08-25 07:41	329.862	20
10024	42353			2014-08-25 07:44	×	1

General Information Probes Map **OpenIPMap Prototype** Download Results

Traceroute results on a geographical map.

OpenIPMap is a prototype visualisation that's attempting to visualise traceroute results geographically. The code is available publicly on GitHub, and the complete project is available separately for those who might want to experiment with it.

OpenIPMap Msms User: BECHA (Xs4all) Help

1733329 www.seil.jp 30prbs 2014-08-25T07:30:00.000Z-2014-08-25TC

- Documentation for analysing measurements results:
 - <https://atlas.ripe.net/docs/rest/>
 - <https://github.com/RIPE-NCC/ripe.atlas.sagan>

- More tools:
 - <https://github.com/RIPE-Atlas-Community>
 - <https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/blob/master/README.md>



Exercise: Analyse Measurement Results



RIPE
NCC

- Get the msm-ID 1004005 (ping IPv6)
 - Measurement data from 2014-09-09 to 2014-09-11
 - <https://atlas.ripe.net/api/v1/measurement/1004005/result/?start=1410220800&stop=1410479999>
- Find out how many times RTT was above 60ms
 - Use Python, Javascript or something else
 - For Javascript, you can use this as a starting point: https://stat.ripe.net/widgets/demo/script_me.html

Save the measurement(s) locally:

```
curl https://atlas.ripe.net/api/v1/measurement/1004005/result/? start=1410220800 > measurement.json
```

Python:

Parse json and find total avg:

```
import json
f = open("measurement.json","r")
measurements = json.load(f)
for m in measurements:
    for r in m["result"]:
        rtt = r["rtt"]
    if rtt > 60: i += 1
i must be > than 14563.
```

Javascript:

```
<script>
var dataAPIUrl = "https://atlas.ripe.net/api/v1/measurement/1004005/result/? start=1410220800";
jQuery.ajax({
url: dataAPIUrl, error: function() {
alert("error"); },
success: function( response ) { var i = 0;
for ( var i = 0, n = response.length; i < n; i++) { var measurement = response[i];
for ( var j = 0, m = measurement.result.length; j < m; j++) { var rtt = measurement.result[j].rtt;
console.log(rtt);
if (rtt > 60)
i++; }
}
jQuery("p").html("The RTT has been above 60ms for " + i + " times");
},
dataType: "jsonp" });
</script>
```

Creating a Measurement



- Log in to atlas.ripe.net
- “My Atlas” > “Measurements”
- Three methods:
 1. Quick & Easy
 - Type
 - Target
 - Done! (default values are used)
 2. Advanced GIU
 3. CLI scripting using API

- Mostly: a periodic, long-term measurement
 - If just want it once, choose “one off”
- Choose type, target, frequency, # of probes, region...
 - Improved, interactive interface helps you at each step
- <https://atlas.ripe.net/doc/udm>
- You will spend credits

- Measurements cost credits
 - Ping = 10 credits, traceroute = 20, etc.
- Why? Fairness and to avoid overload
- Daily spending limit and max number measurements
 - Talk to us if you need more!
- Hosting a RIPE Atlas probe earns credits
 - <https://atlas.ripe.net/doc/credits>
- Earn extra credits by:
 - Being a RIPE NCC member
 - Hosting an anchor
 - Sponsoring probes

- RIPE Atlas <<
- About RIPE Atlas >
- Get Involved >
- Results >
- My Atlas** v

- Probes
- Measurements
- Credits**
- API Keys
- Messages (72 new)
- Anchors
- Sponsorships
- Ambassador Probes
- LIR Benefits
 - Claim 1 Million Credits
 - IPv6 Connectivity Test
 - Quick Look

Account Information

This is where you're able to view the history of your credit use. There are visualisations available, and you can also transfer credits to someone else.

40,900
22,931

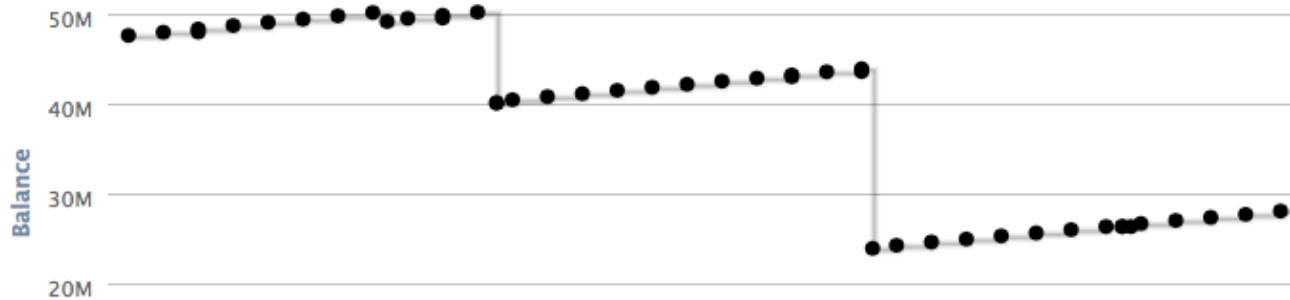
- History
- Charts & Archives
- Transfer

History

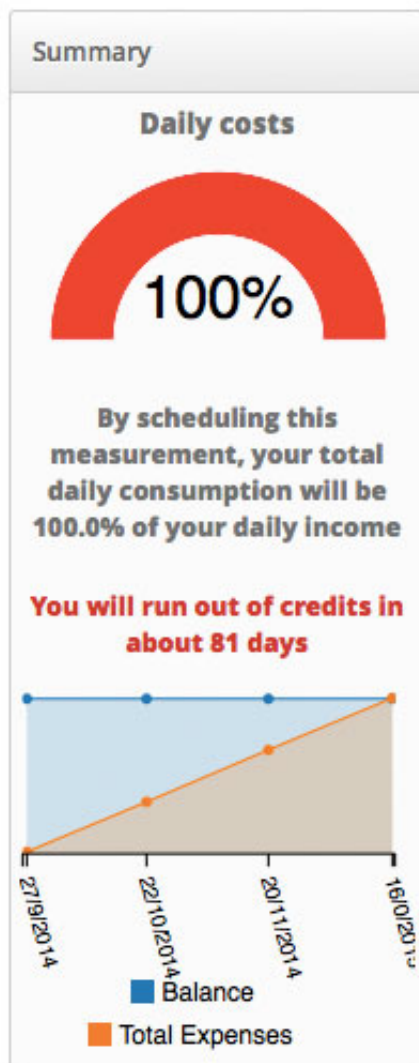
My Atlas / Credits

Give credits to someone

Account Balance
Daily account balance over time



Create a New Measurement BETA



Step 1 Definitions

▼ Ping measurement ✕

<p>Target</p> <input style="width: 90%;" type="text"/> <i>An IP address or hostname</i>	<p>Description</p> <input style="width: 90%;" type="text" value="Ping measurement"/> <i>A free-form description of this measurement</i>
<p>Address Family</p> <input style="width: 90%;" type="text" value="IPv6"/>	<p>Interval</p> <input style="width: 90%;" type="text" value="240"/> <i>How often this should be done (seconds between samples). Note that this value is ignored for one-off measurements.</i>
<p>Packets</p> <input style="width: 90%;" type="text" value="3"/>	<p>Resolve on Probe <input type="checkbox"/></p> <i>Force the probe to do DNS resolution</i>
<p>Size</p> <input style="width: 90%;" type="text" value="48"/>	

+ Ping
+ Traceroute
+ DNS
+ SSL

- <https://atlas.ripe.net/docs/measurement-creation-api/>
- Or copy and paste from GUI
- You will need API keys
 - To create measurements without logging in
 - To securely share your measurement data
- To create, manage and delete API keys:
 - <https://atlas.ripe.net/keys/>
 - <https://atlas.ripe.net/docs/keys/>
- API documentation
 - <https://atlas.ripe.net/docs/measurement-creation-api/>



Exercise: Create a Measurement



- Create a **ping** measurement

- Involving ten probes
- To a target of your choice
- Source: your country
- Duration: two days

1. Create a measurement using the GUI

OR

- 2.1 Create API key

- 2.2 Schedule a measurement using API



Integration with Network Monitoring Systems



- Network operators use tools for monitoring network health
 - For example, Nagios & Icinga
- These tools can receive input from RIPE Atlas via the API
- Benefits:
 - Pings from 500 out of 8,000+ probes around the world
 - See your network from the outside
 - Plug into your existing practices

1. Create a RIPE Atlas ping measurement
2. Go to “status checks” URL
3. Add your alerts in Icinga or Nagios



- Status checks work via RIPE Atlas' RESTful API
 - https://atlas.ripe.net/api/v1/status-checks/MEASUREMENT_ID/
- You define the alert parameters, for example:
 - Threshold for the percentage of probes that successfully received a reply
 - How many most recent measurements to base it on
 - Maximum packet loss acceptable
- Documentation:
 - <https://atlas.ripe.net/docs/status-checks/>

- Community of operators contributed configuration code!
 - Making use of the built-in “check_http” plugin
- GitHub examples:
 - https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/blob/master/scripts_for_nagios_icinga_alerts
- Post on Icinga blog:
 - <https://www.icinga.org/2014/03/05/monitoring-ripe-atlas-status-with-icinga-2/>



Exercise: Set Up “Status Checks”



- Set up and configure a status check:
 - For an existing IPv6 ping msm to www.google.com
 - <https://atlas.ripe.net/api/v1/status-checks/1004005/>
- Configure the status check in such a way that you will trigger an alert for this measurement

- One possible solution:
 - Set the median RTT to a lower level:
 - https://atlas.ripe.net/api/v1/status-checks/1004005/?median_rtt_threshold=10

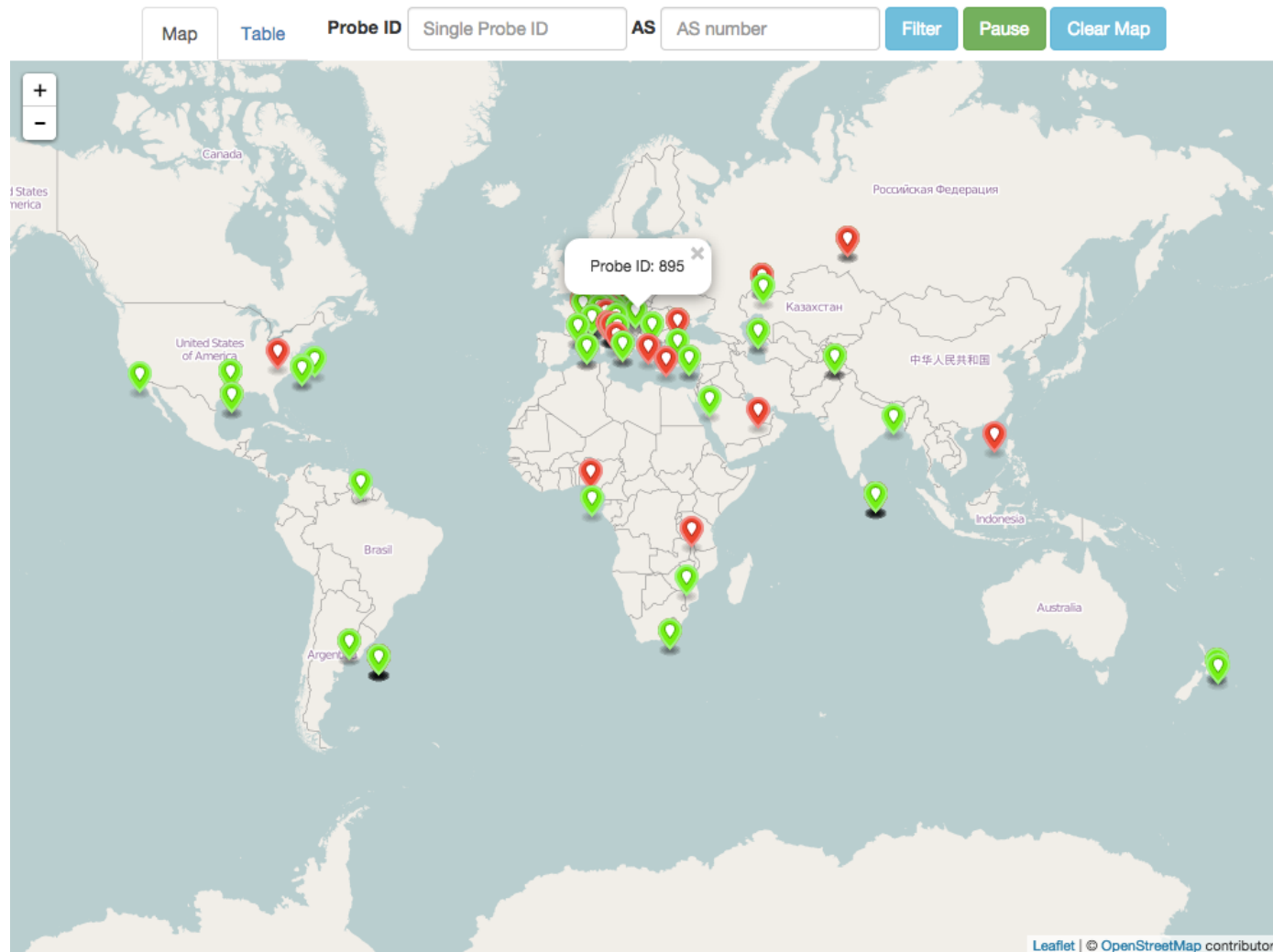


Real-time Performance Monitoring



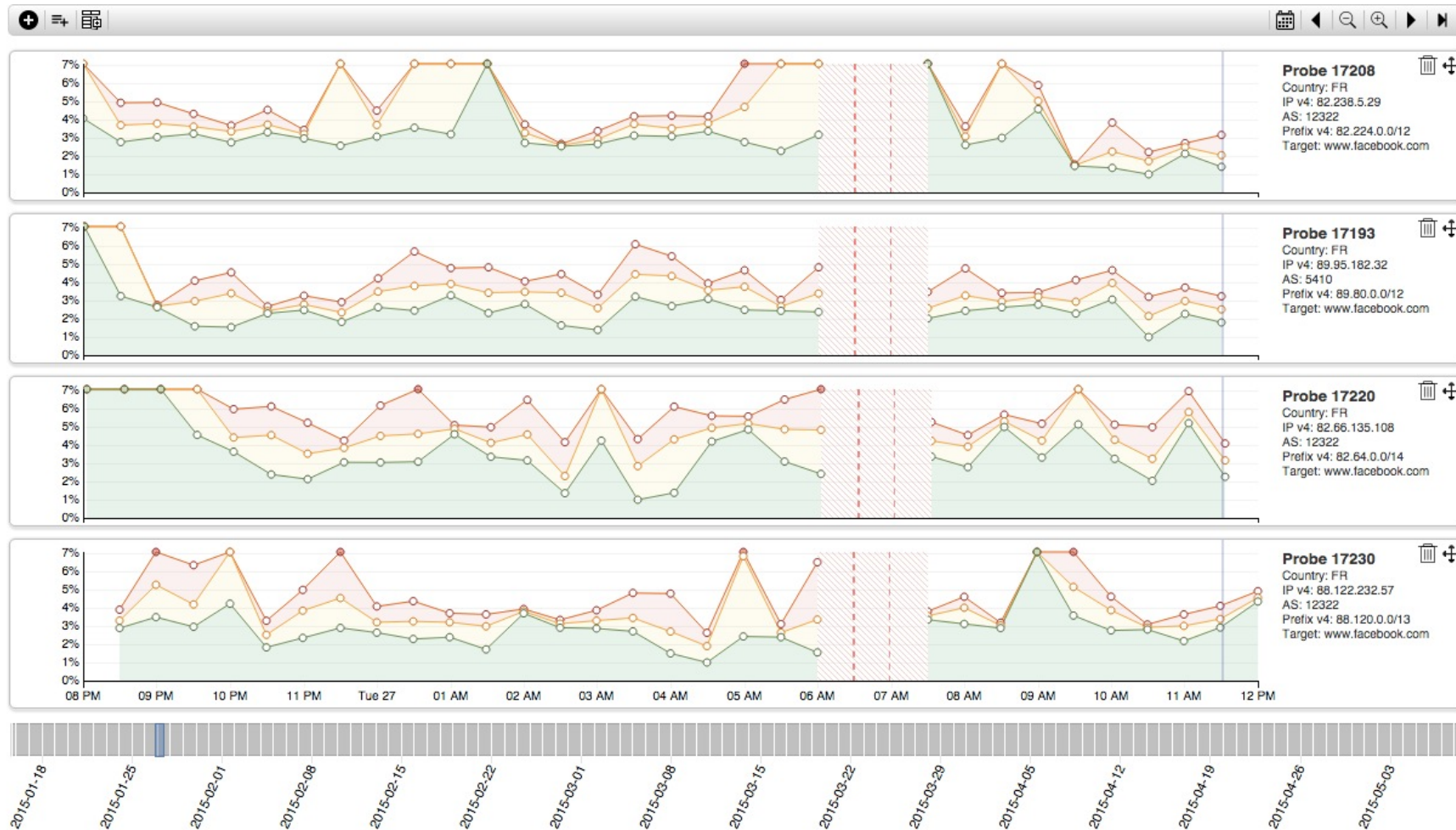
- **RIPE Atlas streaming** is a new architecture that allows users to receive measurement results as soon as they are sent by the probes, **in real time**
 - Publish/subscribe through sockets
- There are two types of data:
 - Measurements results
 - Probe connection status events
- Next: Possibility to replay history

- Visualising network outages
- Server and performance monitoring
- In March 2015: used by almost all hackathon teams:
 - <https://labs.ripe.net/Members/becha/ripe-atlas-hackathon-results>
- Documentation:
 - <https://atlas.ripe.net/docs/result-streaming/>
 - https://labs.ripe.net/Members/suzanne_taylor_muzzin/data-streaming-in-ripe-atlas



- https://labs.ripe.net/Members/andreas_strikos/amsterdam-power-outage-as-seen-by-ripe-atlas

<http://sg-pub.ripe.net/demo-area/massimo-pingviz/>



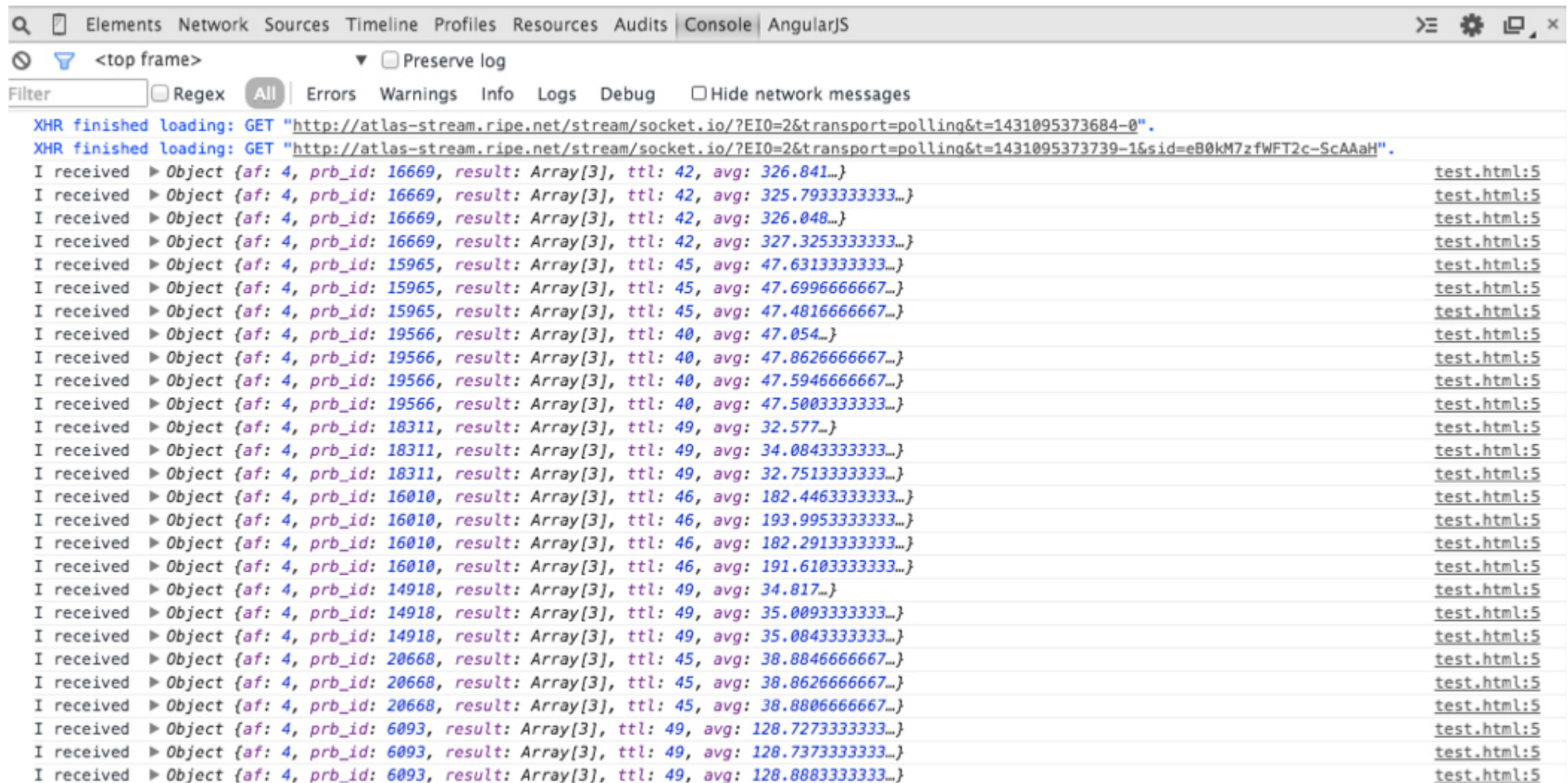


Exercise: Use Streaming API



- Scenario: Customers are occasionally complaining that it takes a long time to reach your service/server
- Action: Ping your server from 500 probes
 - Decide on acceptable latency threshold to apply
 - Notice and react when you start receiving samples
- Task: Use the ping measurement ID1791207
 - Choose the threshold (e.g. greater than 30ms)
 - Impose the threshold on “min” (the minimum result of the three ping attempts)

- Create your empty HTML page
- Copy the example from the documentation:
<https://atlas.ripe.net/docs/result-streaming/>
- Edit the example to fit the task
- Open HTML page
- Open the console
- Wait for results to arrive



The screenshot shows a browser's developer console with the following content:

```
XHR finished loading: GET "http://atlas-stream.ripe.net/stream/socket.io/?EI0=2&transport=polling&t=1431095373684-0".
XHR finished loading: GET "http://atlas-stream.ripe.net/stream/socket.io/?EI0=2&transport=polling&t=1431095373739-1&sid=e80kM7zFWFT2c-ScAAah".
I received ▶ Object {af: 4, prb_id: 16669, result: Array(3), ttl: 42, avg: 326.841...} test.html:5
I received ▶ Object {af: 4, prb_id: 16669, result: Array(3), ttl: 42, avg: 325.7933333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 16669, result: Array(3), ttl: 42, avg: 326.048...} test.html:5
I received ▶ Object {af: 4, prb_id: 16669, result: Array(3), ttl: 42, avg: 327.3253333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 15965, result: Array(3), ttl: 45, avg: 47.6313333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 15965, result: Array(3), ttl: 45, avg: 47.6996666667...} test.html:5
I received ▶ Object {af: 4, prb_id: 15965, result: Array(3), ttl: 45, avg: 47.4816666667...} test.html:5
I received ▶ Object {af: 4, prb_id: 19566, result: Array(3), ttl: 40, avg: 47.054...} test.html:5
I received ▶ Object {af: 4, prb_id: 19566, result: Array(3), ttl: 40, avg: 47.8626666667...} test.html:5
I received ▶ Object {af: 4, prb_id: 19566, result: Array(3), ttl: 40, avg: 47.5946666667...} test.html:5
I received ▶ Object {af: 4, prb_id: 19566, result: Array(3), ttl: 40, avg: 47.5003333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 18311, result: Array(3), ttl: 49, avg: 32.577...} test.html:5
I received ▶ Object {af: 4, prb_id: 18311, result: Array(3), ttl: 49, avg: 34.0843333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 18311, result: Array(3), ttl: 49, avg: 32.7513333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 16010, result: Array(3), ttl: 46, avg: 182.4463333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 16010, result: Array(3), ttl: 46, avg: 193.9953333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 16010, result: Array(3), ttl: 46, avg: 182.2913333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 16010, result: Array(3), ttl: 46, avg: 191.6103333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 14918, result: Array(3), ttl: 49, avg: 34.817...} test.html:5
I received ▶ Object {af: 4, prb_id: 14918, result: Array(3), ttl: 49, avg: 35.0093333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 14918, result: Array(3), ttl: 49, avg: 35.0843333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 20668, result: Array(3), ttl: 45, avg: 38.8846666667...} test.html:5
I received ▶ Object {af: 4, prb_id: 20668, result: Array(3), ttl: 45, avg: 38.8626666667...} test.html:5
I received ▶ Object {af: 4, prb_id: 20668, result: Array(3), ttl: 45, avg: 38.8806666667...} test.html:5
I received ▶ Object {af: 4, prb_id: 6093, result: Array(3), ttl: 49, avg: 128.7273333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 6093, result: Array(3), ttl: 49, avg: 128.7373333333...} test.html:5
I received ▶ Object {af: 4, prb_id: 6093, result: Array(3), ttl: 49, avg: 128.8883333333...} test.html:5
```

```
<script src="http://atlas-stream.ripe.net/socket.io.js"></script>
<script>
  var socket = io("http://atlas-stream.ripe.net:80", { path : "/stream/socket.io" });
  socket.on("atlas_result", function(result){
    console.log("I received ", result);
  });

  socket.emit("atlas_subscribe", {
    stream_type: "result",
    msm: 1791207,
    greaterThan: {min: 30}
  });
</script>
```

- See the connection and disconnection events of all the RIPE Atlas probes in the console of your browser
- Steps:
 - Create your empty HTML page
 - Connect to the streaming API
 - Subscribe to stream_type: “probestatus”

```
<script src="http://atlas-stream.ripe.net/socket.io.js"></script>
<script>
  var socket = io("http://atlas-stream.ripe.net:80", { path : "/stream/socket.io" });

  socket.on("atlas_probestatus", function(status){
    console.log("I received ", status);
  });

  socket.emit("atlas_subscribe", { stream_type: "probestatus" });
</script>
```



- <https://atlas.ripe.net>
- Users' mailing list: ripe-atlas@ripe.net
- Articles and updates on RIPE Labs:
 - <https://labs.ripe.net/atlas>
- Questions and bugs: atlas@ripe.net
- Twitter: [@RIPE_Atlas](https://twitter.com/RIPE_Atlas) and [#RIPEAtlas](https://twitter.com/RIPEAtlas)



Additional Slides

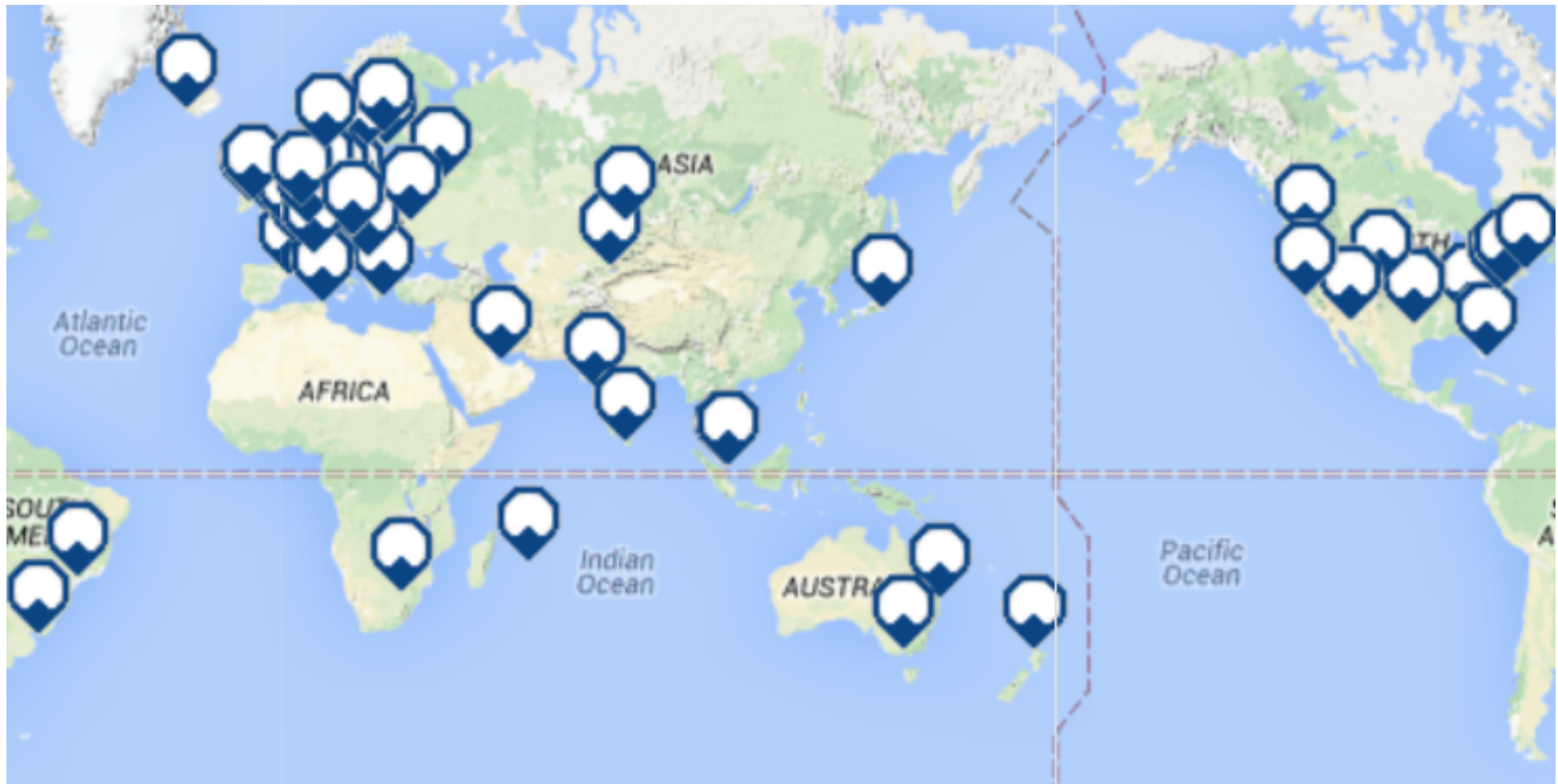


- 8,200+ probes connected
- 5,000+ active users in the last quarter
- 2,500+ results collected per second
- 35,000+ customised measurements weekly
- **Five** types of customised measurements available:
ping, traceroute, DNS, SSL, **NTP** (new)

- For individuals - host a probe:
 - Go to <https://atlas.ripe.net/apply>
 - You will receive a probe by post
 - Register your probe
 - Plug in your probe
 - **One per ASN!**
- For organisations:
 - Host an anchor
 - Sponsor RIPE Atlas
- Help us distribute probes: become an ambassador

- Well-known targets and powerful probes
 - Regional baseline and “future history”
- Anchoring measurements
 - Measurements between anchors
 - 200 probes targeting each anchor with measurements
 - Each probe measures 4-5 anchors
 - Vantage points for DNSMON service
- 120+ RIPE Atlas anchors





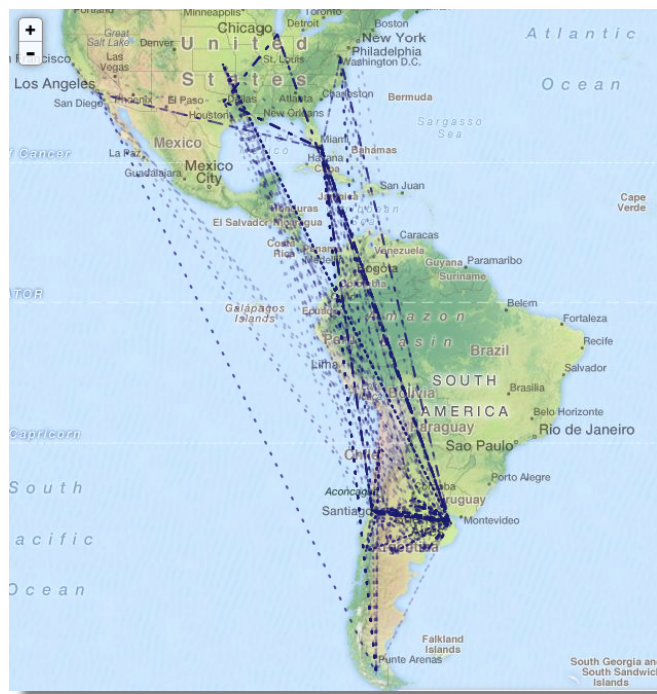
<https://atlas.ripe.net/results/maps/network-coverage/#anchors>

- Impact of IXPs on keeping traffic local

<https://labs.ripe.net/Members/emileaben/measuring-ixps-with-ripe-atlas>

<https://labs.ripe.net/Members/emileaben/measuring-countries-and-ixps-in-the-see-region>

<http://sg-pub.ripe.net/emile/ixp-country-jedi/CL+AR-2015-04/geopath/>



- New measurement type: NTP

https://labs.ripe.net/Members/philip_homburg/ntp-measurements-with-ripe-atlas

Create a New Measurement

Step 1 Definitions

Please select the type of measurement you want to create

+ Ping + Traceroute + DNS + SSL + NTP

Step 2 Probe Selection

Worldwide 50 x

+ New Set - wizard + New Set - manual + IDs List + Reuse a set from a measurement

Step 3 Timing

This is a One-off:

Start time: Stop time:

> Measurement API Compatible Specification

Create My Measurement(s)

- RIPE NCC membership additional benefits:
 - Can perform measurements without hosting a probe
 - Get 1,000,000 credits each month
 - Share probe management with LIR colleagues

- <https://github.com/RIPE-Atlas-Community/openipmap>
- <https://github.com/emileaben/ixp-country-jedi>